République Algérienne Démocratique et Populaire

Ministère de Enseigement Supérieur et de la Recherche Scientifique

Université A.Mira de Béjaia

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de Master

Option : **Administration et Sécurité des Réseaux Informatiques**

Thème

Mise en place d'un IDS en utilisant SNORT
Cas d'étude : CEVITAL

Présenté par :

Mlle.Smahi Imene et Mlle.Tabta Hanane

Devant le jury composé de :

Président	**M. OUZEGANE** Redouane .
Examinatrice	**Mme. BOUADEM** Nassima .
Promoteur	**M. NAFI** Mohammed .

Année universitaire 2021/2022.

Remerciements

$\mathcal{N}$ous tenons d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

$\mathcal{N}$os vifs remerciements aux personnes qui nous ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire.

$\mathcal{N}$ous tenons à remercier notre encadreur Monsieur NAFI mohammed, pour ses directives précieuses, et pour la qualité de son suivi durant la réalisation de ce travail.

$\mathcal{N}$ous remercions très sincèrement, les membres du jury qui ont accepté d'évaluer notre travail. Nous leurs présentons toute notre gratitude et nos profonds respects.

$\mathcal{N}$os remercîments s'adressent également aux enseignants qui ont contribué à notre formation et appuyé notre cursus universitaire, ainsi que le personnel administratif du département d'Informatique.

$\mathcal{N}$ous voulons exprimer nos reconnaissances envers nos amis et camarades qui nous ont apporté leurs supports moraux tout au long de notre travail.

$\mathcal{S}$ans oublier de remercier les personnes du groupe CEVITAL, ARAB Younes, SLIMANI Menad, ALOUI Nagim, pour leur précieux conseils et leurs aides durant toute notre période de stage.

$\mathcal{E}$nfin, nous tenons à remercier toute personne qui a contribué de n'importe qu'elle manière à l'élaboration de ce mémoire.

Dédicaces

$\mathcal{C}$'est avec grand plaisir que je dédie ce modeste travail :

$\mathcal{A}$ ma mère, pour son amour, ses encouragements et ses sacrifices.

$\mathcal{A}$ mon père, pour son soutien, son affection et la confiance qu'il m'a accordé.

$\mathcal{A}$ mon fiancé, pour son encouragement et son soutien qui m'a permis de réaliser le rêve tant attendu.

$\mathcal{A}$ mes chères sœurs $\mathcal{K}$arima, $\mathcal{S}$arah, $\mathcal{A}$mira, $\mathcal{L}$ydia, $\mathcal{I}$mane, $\mathcal{L}$iza, $\mathcal{A}$ssil, et mon frère $\mathcal{T}$ahar, mes sources de joie et de bonheur.

$\mathcal{A}$ mon oncle $\mathcal{J}$ugurta, qui m'a accordé son soutien dans les instants les plus difficilles.

$\mathcal{A}$ toutes personnes de ma grande famille.

$\mathcal{A}$ ma belle famille, $\mathcal{B}$eau père, $\mathcal{B}$elle mère, $\mathcal{R}$iad, $\mathcal{F}$aredj allah, $\mathcal{T}$hilali, $\mathcal{D}$alila, $\mathcal{M}$alika, pour leurs encouragements.

$\mathcal{A}$ mes copines $\mathcal{S}$arah, $\mathcal{M}$elissa, $\mathcal{L}$ynda, $\mathcal{L}$ilia, $\mathcal{N}$adjet, $\mathcal{L}$ynda k.

$\mathcal{A}$ ma chère binôme $\mathcal{I}$mene.

$\mathcal{H}$anane.

Dédicaces

$\mathcal{C}$'est avec grand plaisir que je dédie ce modeste travail :

$\mathcal{A}$ ma mère, pour son amour, ses encouragements et ses sacrifices.

$\mathcal{A}$ mon père, pour son soutien, son affection et la confiance qu'il m'a accordé.

$\mathcal{A}$ mon frères $\mathcal{K}$haled .

$\mathcal{A}$ toute les personnes de ma grande famille.

$\mathcal{A}$ mon amie $\mathcal{M}$arwa et mon ami $\mathcal{A}$thman.

$\mathcal{A}$ ma chère binôme $\mathcal{H}$anane.

$\mathcal{I}$mene.

Table des matières

Table des figures

Liste des Abréviations

ACID : **A**nalyse **C**onsol for **I**ntrusion **D**atabase.
ACK : **AC**quittement.
ADODB : **A**ctive **D**ata **O**bjet **D**ata **B**ase.
B.A.S.E : **B**asic **A**nalysis **S**ecurity **E**ngine.
DMZ : **D**e**M**ilitarized **Z**one.
DNS : **D**omain **N**ame **S**ervice.
DSI : **D**irection **S**ystem **I**nformation.
FTP : **F**ile **T**ransfer **P**rotocol.
GCC : **GNU** **C**ompiler **C**ollection.
HIDS : **H**ost **I**ntrusion **D**etection **S**ystem.
HTML : **H**yper **T**ext **M**arkup **L**anguage.
HTTPS : **H**yper**T**ext **T**ransfer **P**rotocol **S**ecure.
ICMP : **I**nternet **C**ontrol **M**essage Protocol.
ID : **ID**entifiant.
IDS : **I**ntrusion **D**etection **S**ystem.
IDMEF : **I**ntrusion **D**etection **M**essage **E**xchange **F**ormat.
IP : **I**nternet **P**rotocol.
Libnet : **lib**rairie **N**etwork.
Libpcap : **Lib**rairie **P**acket **CAP**ture.
Libpcre : **lib**rairie **P**erl **C**ompatible **R**egular **E**xpression.
NIDS : **N**etwork **B**ased **I**ntrusion **D**etection **S**ystem.
NMAP : **N**etwork **MAP**per.
PHP : **P**reprocessor **H**y**P**ertext.
SEQ : **SEQ**uence.
SGBD : **S**ystème de **G**estion de **B**ase de **D**onnée.
SMB : **S**erver **M**essage **B**lock.
SNMP : **S**imple **N**etwork **M**anagement **P**rotocol.
SSH : **S**ecure **SH**ell.
TCP : **T**ransmission **C**ontrol **P**rotocol.
TOS : **T**aux **O**ndes **S**tationnaires.
TTL : **T**ime **T**o **L**ive.
UDP : **U**ser **D**atagram **P**rotocol.
XML : e**X**tensiblee **M**arkup **L**anguag.

Introduction générale

Les réseaux informatiques sont plus importants qu'ils ne l'étaient il y a quelques années. Aujourd'hui, les entreprises n'hésitent pas à créer des réseaux informatiques pour faciliter la gestion des infrastructures. Par conséquent, la sécurité de ces réseaux est un enjeu important.
La sécurité informatique est une grande préoccupation dans la gestion des réseaux d'entreprise ainsi que des réseaux personnels.
Divers mécanismes ont été introduits pour résoudre ces problèmes de sécurité, tels que des pare-feu et des programmes antivirus, mais le développement rapide des techniques de piratage les ont poussés à leurs limites. Pour éviter ces restrictions, il est indispensable d'utiliser un système de détections d'intrusions. Un système de détection d'intrusions est un ensemble de méthodes et de techniques utilisées pour détecter les activités suspectes, au niveau du réseau et ou niveau de l'hôte.
L'objectif de notre travail est de mettre en place SNORT, qui est un système de détection d'intrusion réseau (NIDS) open source, fonctionnant sur les systèmes Windows et linux. il est capable d'effectuer en temps réel des analyses de trafic et de loger les paquets sur un réseau IP.
Ce mémoire est structuré en quatre chapitres :
Dans le premier chapitre, nous présentons les généralités sur le système de détection de'intrusions en donnant leurs différents types et le principe de son fonctionnement.
Le second chapitre est consacré à la présentation des généralités sur SNORT.
Dans le troisième chapitre, nous présentons l'entreprise accueill CEVITAL, son activité, ses différentes directions.
Le quatrième chapitre est consacré à la mise en place d'un IDS en utilisant SNORT, notamment le choix de la bonne architecture, et la mise en place d'une solution et les tests.
Enfin, nous terminons le mémoire par une conclusion générale et quelques perspectives.

Chapitre 1

Généralités sur les systèmes de détection d'intrusions

Introduction

Aujourd'hui, le monde connaît des avancées majeures en informatique. Les besoins en équipements (matériels ou logiciels) de sécurité sont devenir plus en demande.
Le monde continue de faire évoluer les technologies conçues pour protéger les données contre les attaques internes ou externes, jour après jour. Plusieurs inventions sont réalisées telles que les anti-virus, le pare-feu et anti-spam,etc.
Malheureusement, les systèmes antivirus ou pare-feu sont généralement inefficaces contre certaines menaces. Ce qui a permis d'inventer un système plus performant et très efficace contre les attaques : appelé Système de détection d'intrusions(IDS).
Dans ce chapitre nous avons présenté tout d'abord la notion de système de détection d'intrusion ainsi son architecture, et nous avons présenté ses trois types.

1.1 Définition d'un IDS

Un système de détection d'intrusions (IDS) est un programme matériel ou logiciel qui permet de détecter et d'analyser les attaques et les logiciels malveillants en analysant le trafic selon des règles. [1] Certains termes sont souvent utilisés quand on parle d'IDS qui sont : le faux positif et faux négatif
– **Faux positif :** une alerte remontée par un IDS, mais qui ne correspond pas à une attaque réelle.
– **Faux négatif :** une intrusion réelle qui n'a pas été détectée par l'IDS.

1.2 Architecture d'un IDS

Plusieurs architectures ont été proposées pour décrire les différents éléments composent un système de détection d'intrusions. L'architecture la plus simple est composée de trois modules : la source des données, l'analyseur des données et le module des réponses [2], comme le montre la figure 1.1. [3]

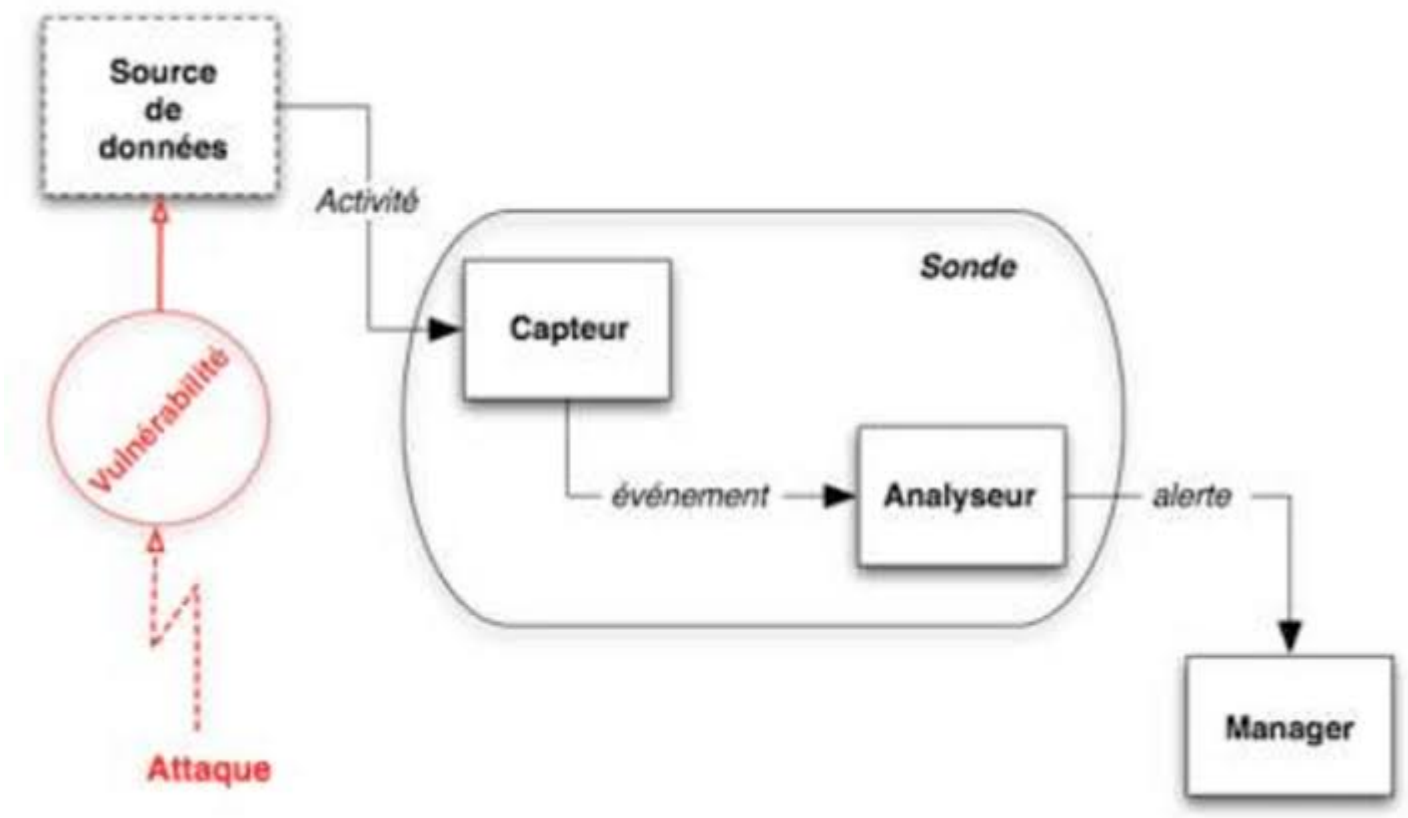

FIGURE 1.1 – Architecture d'un IDS.

1.2.1 Source des données

Les sources de données, également appelées sondes de capture, sont chargées de récupérer les informations et événements liés à la détection et de les envoyer au module d'analyse. L'emplacement de la sonde de capture joue un rôle très important dans la qualité de la détection. Les captures peuvent transmettre ces données brutes directement, mais généralement avec un prétraitement. Selon la source de données utilisée pour observer l'activité du système, il existe trois types de captures : les captures système, les captures réseau et les captures d'application.

1.2.2 Analyseur des données

C'est le cœur d'IDS, ce module permet d'analyser les informations collectées par les sondes de capture. Il utilise une base de connaissances liée aux attaques, pour la recherche des activités malveillantes.

1.2.3 Module de réponse

Le module de réponse, appelé aussi le manager, collecte les alertes produites par le capteur. Il assure les réponses d 'IDS aux activités malveillantes détectées. Les réponses peuvent être actives ou passives, c'est les contre-mesures nécessaires pour contrer les intrusions. Ça peut être un simple message d'alerte, une sauvegarde dans un fichier log ou une interruption d'une connexion.

1.3 Caractéristique d'un IDS

Pour mener à bien ses taches, un système de détection d'intrusion doit vérifier certaines caractéristiques liées à ses fonctionnalités qui sont : [2]

1.3.1 Exactitude

Elle représente la concordance maximale des résultats de l'IDS avec le comportement normal du système surveillé. L'IDS doit connaitre parfaitement le fonctionnement du système et ne pas le confondre avec des activités intrusives. Cette caractéristique peut être traduite par un taux de faux positifs minimal.

1.3.2 Temps de réponse

Il s'agit de la vitesse de traitement des événements, qui doit être maximisée pour éviter les retards et permettre une détection en temps réel. L'IDS doit également être en mesure de diffuser rapidement les résultats de détection aux administrateurs système et/ou de prendre des contre-mesures dans un court laps de temps.

1.3.3 Exhaustivité de détection

Un IDS idéal devrait être capable de détecter toutes les attaques connues et inconnues. Cette mesure est difficile à évaluer par manque de bonne connaissance de l'attaque.

1.3.4 Tolérance aux fautes

Le système de détection d'intrusion lui-même doit résister aux attaques. Ceci permet d'éviter toute tentative d'outrepasser l'IDS.

1.4 Fonctionnement d'un IDS

Deux aspects dans le fonctionnement d'un IDS peut être distingués : le mode de détection utilisé et la réponse apportée lors de la détection d'une intrusion. Il existe deux modes de détections : la détection d'anomalies et la reconnaissance de signatures. Deux types de réponses existent : la réponse passive et la réponse active. [4]

1.4.1 Modes de détection

- **Détection d'anomalies**
  la détection d'anomalies liées au profil de trafic général. Les implémentations comportent toujours une phase d'apprentissage au cours de laquelle l'IDS va découvrir le fonctionnement normal de l'élément surveillé. Ainsi, pu signaler des écarts par rapport à l'opération de référence. Des modèles comportementaux peuvent être développés à partir d'analyses statistiques. Ils ont l'avantage de détecter de nouveaux types d'attaques. Cependant, des ajustements fréquents sont nécessaires pour améliorer le modèle de référence afin qu'il reflète l'activité normale de l'utilisateur et réduise le nombre de fausses alertes générées. Pour HIDS (hote intrusion detection system) ,une telle détection peut être basée sur des informations telles que l'utilisation du processeur, l'activité du disque, le temps de connexion ou l'utilisation de certains fichiers.

– **Détection et reconnaissance de signatures**
Cette approche consiste à rechercher dans l'activité des éléments surveillés des empreintes digitales (ou signatures) d'attaques connues. Ce type d'IDS est purement réactif, il ne peut détecter que les attaques avec signatures. Par conséquent, il doit être mis à jour fréquemment. De plus, l'efficacité de ce système de détection dépend en grande partie de la précision de sa bibliothèque de fonctionnalités. C'est pourquoi ces systèmes sont contournés par les pirates, qui utilisent des techniques dites "d'évasion", notamment en camouflant les attaques utilisées. Ces techniques ont tendance à modifier les caractéristiques d'une attaque, de sorte qu'IDS ne reconnaît plus ces caractéristiques. Des signatures plus générales peuvent être développées, permettant de détecter des variantes d'une même attaque, mais cela nécessite une bonne compréhension de l'attaque et du réseau. Les attaques peuvent être caractérisées au niveau des paquets (jusqu'à TCP ou UDP) ou au niveau du protocole (HTTP, FTP) afin de bloquer les variantes de l'attaque et de ne pas interférer avec les signatures normales du trafic réseau. Au niveau des paquets, l'IDS analyse tous les paquets passants et les compare avec les caractéristiques des attaques connues. Au niveau du protocole, l'IDS vérifiere si la commande envoyée est correcte ou ne contient pas d'attaque. Cette fonctionnalité est principalement développée pour http.

1.4.2 Mode de réponse

Si l'IDS détecte une attaque, deux comportements peuvent être adoptés : une réponse passive ou bien une réponse active. [3]

– **Réponse passive**
La plupart des systèmes de détection d'intrusion ne fournissent que des réponses passives aux intrusions, et lorsqu'une attaque est détectée, ils génèrent des alertes et notifient les administrateurs système par e-mail, ou messages. À ce moment-là, ce dernier devra prendre les mesures nécessaires.
– **Réponse active**
En plus de notifier les opérateurs, d'autres systèmes de détection d'intrusion peuvent automatiquement prendre des mesures pour arrêter une attaque en cours. Par exemple, ils peuvent couper les connexions suspectes ou même reconfigurer les pare-feu pour empêcher tout accès aux sites incriminés en réponse à des attaques externes. Des outils tels que Real Secure [Sys98] ou NetProwler6 permettent ce type de réaction. Cependant, ce type de fonctionnalité automatisée semble être potentiellement dangereux, car il pourrait conduire à un déni de service causé par IDS. Par exemple, un attaquant déterminé pourrait usurper un IDS en usurpant des adresses de réseau local, qui seraient alors considérées par l'IDS comme la source de l'attaque. Il est préférable de fournir des réponses facultatives à l'opérateur humain (qui prend la décision finale).

1.5 Choix du placement d'un IDS

L'emplacement de l'IDS dépendra des politiques de sécurité définies dans le réseau. Il serait intéressant de placer des IDS : [5]
La figure 1.2 illustre les différents emplacement d'un IDS [6]

– Dans la zone démilitarisée (attaques contre les systèmes publics).
– Dans le (ou les) réseau(x) privé(s) (intrusions vers ou depuis le réseau interne).
– Sur la patte extérieure du firewall (détection de signes d'attaques parmi tout le trafic entrant et sortant, avant que n'importe quelle protection intervienne).

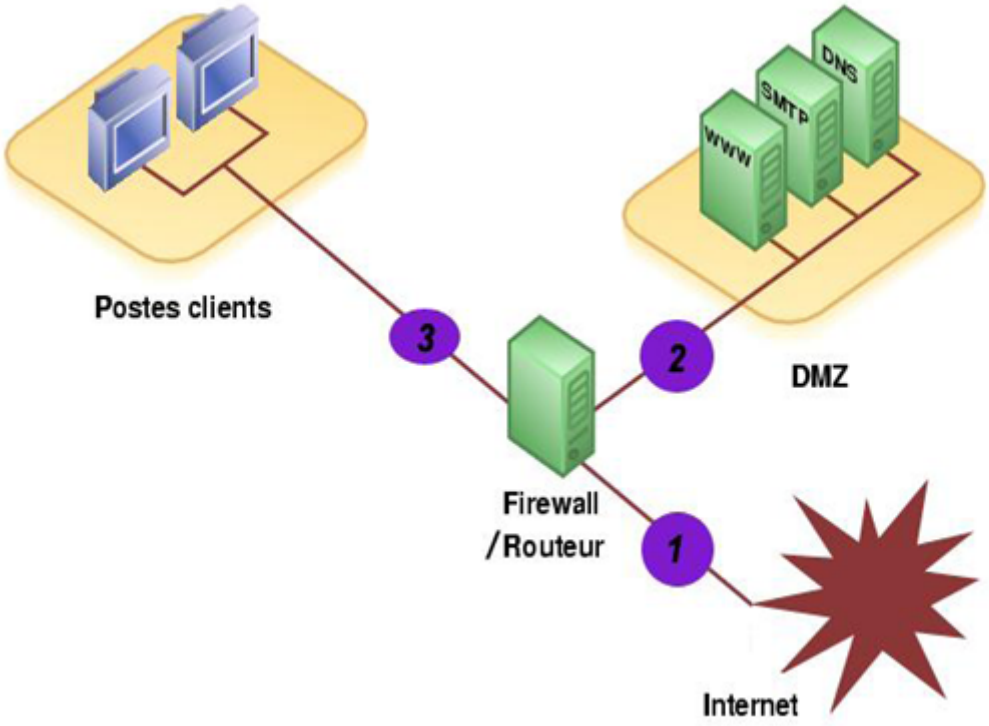

FIGURE 1.2 – Placement d'IDS.

1.6 Types de système détection d'intrusion

Il existe 3 grandes familles d'IDS qui sont : HIDS, NIDS et Hybride

1.6.1 HIDS (Système Détection Intrusion Hôte)

Un HIDS est un système de détection basé sur l'hôte, installe généralement sur une machine pour protéger et analyser les flux et les activités.
HIDS surveille la politique de sécurité du système contre toute tentative de piratage interne ou externe. Les systèmes de détection peuvent consulter les fichiers journaux du système et des applications pour détecter toute activité d'intrusions. Certains systèmes sont réactifs, ce qui signifie qu'ils informent uniquement lorsque quelque chose arrive. D'autres sont proactifs, ils peuvent renifler le trafic réseau à un particulier. [7]
La machine peut être surveillée sur plusieurs points : [8]

- **Activité de la machine :** nombre et liste de processus ainsi que d'utilisateurs. et ressources consommées.
- **Activité de l'utilisateur :** horaire et durée de connexions, commandes utilisées, messages envoyés, programmes activés (dépassement du périmètre défini.
- **Activité malicieuse :** d'un ver, Virus ou cheval de Troie, etc, tourné la politique de sécurité du système.

Les HIDS utilisant deux types de sources pour fournir des informations sur l'activité : les logs et les traces. La figure 1.3 montre l'architecture d'un HIDS. [9]

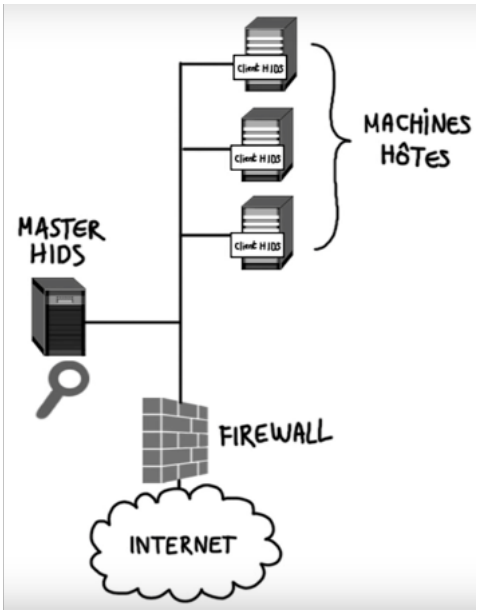

FIGURE 1.3 – Schéma d'architecture HIDS.

Fonctionnalités d'un HIDS

Les HIDS possèdent toutes ou une partie des fonctionnalités suivantes : [10]

- **Surveillance des fichiers journaux**
  Les fichiers journaux sont surveillés de plusieurs façons afin qu'ils soient d'abord exécutés logcheck a tous les heures et après chaque redémarrage. Il analyse les modèles dans les journaux (par exemple : syslog et auth.log).
  Selon le niveau de configuration, nous pouvons définir la vérification peut être configuré à ce niveau : Paranoïaque, Serveur ou Poste de travail, par ordre décroissant de détail. une autre fois, le moteur OSSEC fonctionne sur la base de schémas décrits dans des fichiers XML. Dans la règle décrite, il est possible de spécifier qu'une alerte ne doit être déclenchée que si la cause a été préalablement déterminée dans un délai défini.
- **Contrôles d'intégrité des fichiers**
  Contrôle d'intégrité basé sur des fonctions de hachage cryptographiques. Ce dernier permet à partir d'une entrée de taille variable (par exemple, un fichier), de produire une chaîne d'octets de taille fixe, dont la longueur dépend de la fonctionutilisée. Elle a de plus les propriétés suivantes : risistance à la pré-image, résistante à la seconde pré-image, résistance aux collisions.
- **Détection de rootkits**
  Un rootkit est un ensemble de modifications apportées par des attaquants pour masquer leurs traces. Les rootkits peuvent être à plusieurs niveaux : binaire (exécutables et bibliothèques) ou noyau. Les rootkits binaires affectent souvent des outils comme ps, ls, kill, top, du, find, netstat, etc. et des bibliothèques, en particulier des bibliothèques dynamiques telles que libproc.so. Ces fichiers sont modifiés pour masquer l'existence de fichiers ou de processus appartenant à des attaquants du système. Toutes ces modifications peuvent être détectées par des vérifications d'intégrité, et Les rootkits du noyau fonctionnent après avoir modifié la source du noyau et recompilé (de manière fastidieuse), ou plus difficilement, en écrivant sur un périphérique tel que /dev/kmem Les opérations de détection de rootkit du noyau recherchent généralement des modèles connus dans les modules chargés et vérifient que les appels système n'ont pas été modifiés.

– **Détection de comportements douteux**
La détection d'un comportement suspect consiste à partir d'un constat simple : un programme est conçu pour effectuer un ensemble défini de tâches. Si un processus exécute une tâche différente de celle prévue, il s'écarte de son comportement normal, ce qui présente un risque pour le système. Il est donc nécessaire de suivre l'évolution du procédé et de s'assurer qu'il respecte la finalité pour laquelle il a été conçu. Cela peut être fait de plusieurs façons. Par exemple, Systrace permet de définir des politiques de sécurité au niveau des appels système.

Voici quelques exemples de HIDS :

– Chkroottkit.
– DarkSpy.
– FChek.
– Integrit.
– OSSEC.
– Osiri.

1.6.2 NIDS (Système Détection Intrusion Réseau)

Les NIDS sont des systèmes de détection d'intrusions qui capturent les paquets de données circulant sur le support réseau et les faire le correspondre à une base de données de signatures. Selon qu'un paquet correspond à une signature d'intrus, une alerte est générée ou le paquet est enregistré dans un fichier ou base de données. NIDS nécessitent généralement un accès réseau en mode promiscuité afin d'analyser tout le trafic.
Les NIDS sont des dispositifs passifs qui n'interfèrent pas avec le trafic qu'ils surveillent.
Le NIDS renifle l'interface interne du pare-feu en mode lecture seule et envoie des alertes à un serveur de gestion NIDS via une interface réseau différente (c'est-à-dire en lecture/écriture). [7]
Figure 1.4 :Montre une architecture d'un NIDS. [9]

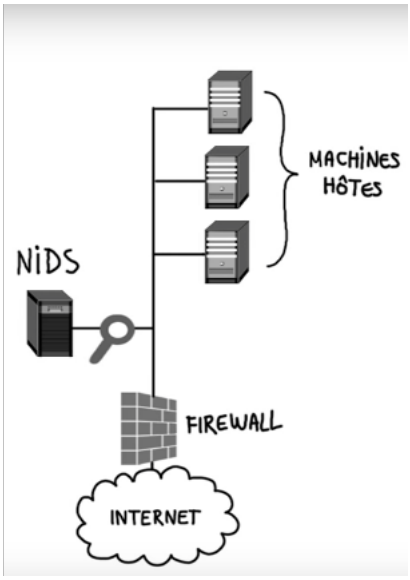

FIGURE 1.4 – Schéma d'achitecture d'un NIDS.

1. **Partie d'un IDS**

   Un NIDS est compose en quatre parties : la capture, les signatures, l'analyse et les alertes [8], comme le montre la figure1.5.

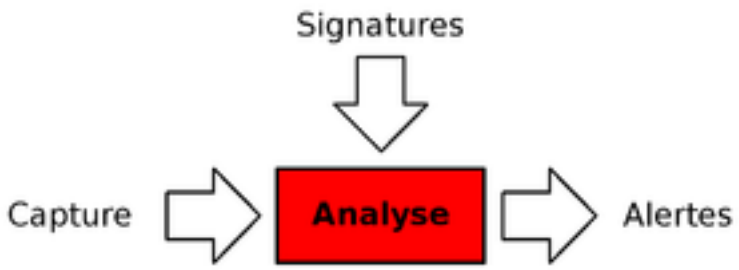

FIGURE 1.5 – Partie d'un NIDS.

   (a) **Capture**

   La capture est utilisée pour récupérer le trafic réseau. habituellement en temps réel. bien que certains NIDS permettent l'analyse du trafic capturé. La plupart des NIDS utilisent des bibliothèques de capture de paquets standards libpcap. Par conséquent, les fonctions capturées par NIDS sont généralement étroitement liées à cette libpcap en utilisant l'interface : pcap et winpcap

   Pcap : "paquet capture" pour capturer le trafic réseau.

   Winpcap : sous Windows.

   Il fonctionne en copiant tout paquet arrivant au niveau couche liaison de données. Certains paquets peuvent être abandonnés car sous une forte charge.

   (b) **Signature**

   Les bibliothèques de signatures rendent l'approche d'analyse similaire à celle de l'antivirus car elles sont basées sur les attaques. Par conséquent, NIDS est efficace s'il connaît l'attaque. Des outils gratuits ou commerciaux ont évolué pour fournir une personnalisation de signature en réponse à des attaques.

   Les outils basés sur les signatures nécessitent des mises à jour très fréquentes.

   NIDS à l'avantage d'être un système a la capacité de détecter les attaques qui ciblent la machine à la fois. Leurs limites sont le taux élevé de faux positifs qu'il génère, le fait que les signatures sont toujours vulnérables aux attaques et qu'elles peuvent être la cible d'une attaque.

   (c) **Analyse**

   A partir des éléments donnés dans l'introduction, le moteur d'analyse met ces éléments de relation en employant plusieurs techniques : la refragmentation, la dissection protocolaire.

   – **Refragmentation**

     Les NIDS ont le devoir de refragmenter les paquets avant analyse, afin de ne pas manquer une attaque. Il s'agit d'une opération relativement complexe, étant donné que chaque hôte de destination ne refragmente pas de la même façon, selon le système d'exploitation sur lequel l'attaque est visée. Il s'agit encore d'une technique d'évasion utilisable aujourd'hui car les NIDS ne sont pas forcément configurés correctement pour gérer un cas précis.

   – **Dissection protocolaire**

     La dissection permet de comprendre un protocole donné, et de le décoder pour l'analyser. C'est la partie la plus sensible du NIDS parce qu'elle correspond au plus grand vecteur d'attaque.

(d) **Alertes**

Les alertes sont généralement stockées dans le syslog. Cependant il existe une norme qui permet d'en formaliser le contenu, afin de permettre à différents éléments de sécurité d'inter opérer. Ce format s'appelle IDMEF (pour Intrusion Détection Message Exchange Format) décrit dans la RFC4765.

2. **Principales techniques d'un NIDS**

   Pour détecter des intrusions les NIDS implémentent principalement les techniques suivantes : [1]

   - **Le pattern matching :** qui consiste à réparer des chaines de caractères bien identifiées liées à des séquences d'attaque. les systèmes antivirus implémentent des algorithmes similaires. Cette méthode est relativement fiable mais nécessite que les attaques aient été identifiées et codées sous forme de signatures au préalable.
   - **L'analyse de protocoles :** qui consiste à analyser la structure des paquets et l'utilisation de certains paramètres non conformes aux normes officielles. Elle a tendance à déclencher un peu plus de fausses alertes que le pattern matching.
   - **La détection d'anomalies :** c'est le terme que l'on trouve souvent dans littérature consacré aux IDS mais qui n'a pas forcément la même définition d'un éditeur à l'autre. La détection d'anomalies consiste à détecter toute déviation par rapport à un modèle correspondant à un comportement normal.

Parmi les NIDS, nous pouvons citer : NetRanger, Dragon, NFR, DTK, BRO, SNORT.

1.6.3 IDS Hybride (Système Détection Intrusion Hybride)

Les systèmes de détection d'intrusion hybrides combinent les propriétés de plusieurs systèmes de détection différents. En pratique, la combinaison de NIDS et HIDS permet de surveiller les réseaux et les hôtes. Souvent utilisés dans des environnements décentralisés, ils peuvent recueillir des informations auprès de diverses sondes placées sur le réseau. Leur nom "Hybride" vient du fait qu'ils sont capables de collecter à la fois des informations du système HIDS que NIDS. [4]
Cet IDS permet de stocker dans une base de données des alertes provenant de différents systèmes.
Utilisant SNORT comme NIDS, et d'autres logiciels en tant que HIDS, il permet de combiner des outils puissants tous ensemble pour permettre une visualisation centralisée des attaques.
Les IDS hybrides sont basés sur une architecture distribuée, où chaque composant unifie son format d'envoi d'alerte (typiquement IDMEF) permettant à des composants divers de communiquer et d'extraire des alertes plus pertinentes, comme le montre la figure 1.6.

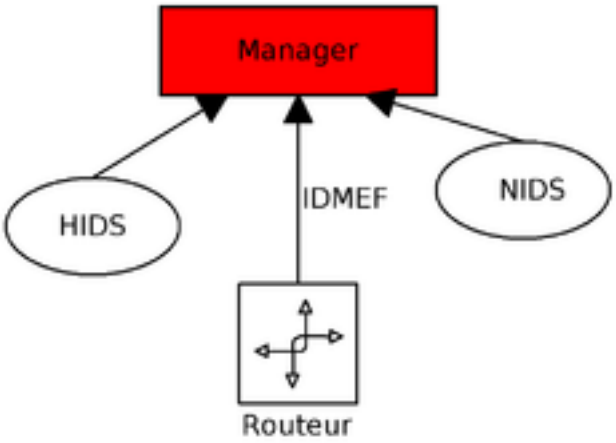

FIGURE 1.6 – Architecture d'un IDS Hybride.

Les IDS hybrides se décomposent en deux parties : La corrélation, et l'harmonisation des formats. [8]

1. **Corrélation**
   La corrélation est la connexion entre deux ou plusieurs scènes laquelle de ces éléments crée ou affecte. Elle se traduit plus généralement par la transformation d'une ou plusieurs alertes en attaque. Cela facilite la compréhension des attaques au lieu d'être dispersé parmi les alertes. Idéalement, il nécessite un IDS hybrides car plus les informations sur un événement sont hétérogènes, plus la corrélation est pertinente.
   Les formats ont été normalisés (IDMEF), il ne reste plus qu'à faire le lien pour détecter les alertes qui n'arriveront jamais sur l'analyseur seul. Il y a deux corrélations : corrélation passive et active :
   - **Corrélation passive** Correspond à une génération d'alerte basée sur celles existantes. Nous pouvons prendre par exemple les scans de Force brute SSH.
   - **Corrélation active** Qui va chercher les informations correspondant à des alertes émises. Par exemple, lorsqu'une personne se connecte en dehors des heures de travail, ceci a un impact élevé.
2. **Harmonisation des formats**
   Le format IDMEF décrit complètement une alerte. Une alerte correspond à un message envoyé depuis un analyseur, qui est une sonde en langage IDMEF, vers un récepteur.
   Le but d'IDMEF est de proposer une norme qui permette la communication hétérogène de tout l'environnement ou les capacités d'un analyseur donné. IDMEF fournit également un vocabulaire précis, qui champ de détection d'intrusion. Par exemple, un classement correspond au nom de l'alerte, l'impact du niveau d'attaque.

Quelques exemples d'IDS hybride sont :
- Prelude
- OSSIM

1.7 Avantages et les inconvénients d'un IDS

L'IDS présentent plusieurs avantages et inconvénients qui sont enumeré dans ce qui suit :

1.7.1 Avantages d'un IDS

- Les IDS peuvent analyser tout le trafic dans le même domaine de collision, et relever des attaques, alors même qu'ils ne sont pas la cible directe.
- Il peut être intéressant de positionner les sondes pour étudier l'efficacité des protections mises en place.
- Les IDS se cantonnent à la surveillance des systèmes sur lesquels ils sont hébergés.
- Ils contient des outils de filtrage très intéressants qui nous permettent de faire du contrôle par protocole (ICMP, TCP, UDP), adresse IP, suivi de connexion.
- Ils diminuent le travail manuel de la sécurité, en réduisant le coût dans les entreprises.
- Ils se protéger des attaques passant par des flux autorisés.

1.7.2 Inconvénients d'un IDS

- La configuration, et l'administration des IDS nécessitent beaucoup de temps, et de connaissances.
- La faiblesse d'un IDS est liée à la faiblesse de la plate-forme.
- Une saturation de la mémoire, de la carte réseau, ou du processeur porte atteinte directement au bon fonctionnement de tout le système.
- Les interfaces fournissent beaucoup d'informations, et permettent des tris facilitant beaucoup le travail, mais l'intervention humaine est toujours indispensable, pour prendre les décisions.
- Problème de positionnement des sondes se pose. La mise en place est importante.

Conclusion

Les IDS jouent un rôle très important dans le processus de protection des systèmes d'information. Ils sont très efficace dans la détection des activités malveillantes et des tentatives d'outrepasser les autres mécanismes de sécurité.
Dans ce chapitre, nous avons expliqué la notion de système de détection d'intrusions avec ces différents types, son architecture, ainsi que son fonctionnement.
Dans le chapitre suivant, nous allons particulierement étudier un système de détection d'intrusion réseau appelé Snort, Ce dernier est l'un des plus populaires dans le monde de détection d'intrusion.

Chapitre 2

Généralités sur SNORT

Introduction

SNORT est sans doute l'IDS le plus célèbre dans le monde du logiciel libre. Il existe de nombreux projets intéressants liés à la détection d'intrusion dans le monde open source, tels que Prelude, mais aucun n'a jusqu'à présent été aussi notoire que SNORT.
SNORT a été développé pour la première fois en novembre 1998 par Martin Roesh. Il était à l'origine destiné à être utilisé comme renifleur de paquets. Depuis, c'est devenu de plus en plus. Il est actuellement utilisé dans la plupart des situations IDS. Il a été porté sur diverses plates-formes actuellement SNORT est sur Windows, FreeBSD, Linux et Solaris.
Dans ce qui suit, nous allons présenté en générale SNORT.

2.1 Définition de SNORT

SNORT est un système de détection d'intrusion réseau (NIDS) open source disponible gratuitement. Il est capable d'analyser le trafic en temps réel. Il utilise les règles pour trouver des paquets correspondants et générer des alertes pour les utilisateurs. [11]
SNORT a trois utilisations principales :
– En tant que renifleur de paquets comme Tcpdump.
– En tant qu'enregistreur de paquets ce qui est utile pour le débogage du trafic réseau.
– Un système complet de prévention des intrusions sur le réseau.

2.2 Règles de SNORT

Chaque règle de SNORT doivent être écrites sur une seule ligne, car l'analyseur ne sait pas comment traiter la règle sur plusieurs lignes. Les règles de SNORT sont divisées en deux sections : [1]
– La section d'en-tête contient l'action liée à la règle et des informations sur les adresses IP source et de destination, les masques de sous-réseau et les ports concernés.
– La section des options contient le message d'alerte et des informations sur la façon dont le paquet doit être inspecté et si l'action spécifiée par l'en-tête doit être prise.
La figure 2.1 montre un exemple de règle de SNORT.

FIGURE 2.1 – Exemple règle de SNORT.

2.2.1 Partie en-tête

1. **Action de règles**
   Les règles Snort décrivent une action de base. Il y a plusieurs actions possibles à effectuer sur le trafic :
   Alert : Génère une alerte et enregistre le paquet.
   Log : Enregistre le paquet sans générer d'alertes.
   Pass : Ignore le paquet.
   Activate : Génère une alerte et active une règle qui était dormante.
   Dynamic : Reste inactif tant qu'une règle n'a pas utilisé la directive activate puis enregistre les paquets incriminés.
2. **Porotocoles**
   La règle doit indiquer quel protocole surveiller. En effet, observer le trafic pour tous les protocoles est très difficile et très lourd en termes de bande passante, c'est pourquoi un protocole est specifie pour chaque règle. [12]
   Voici quelques choix des protocoles :
   - **TCP :** Est un protocole de la couche de transport, orienté connexion, conçu pour fournir une connexion fiable pour l'échange de données entre deux systèmes. TCP garantit que tous les paquets sont correctement séquencés et reconnus, et qu'une conversation est établie avant l'envoi des données. Cela garantit que les deux machines sont prêtes à communiquer et que les informations transmises d'un système à l'autre sont acheminées sans perte.
   - **UDP :** Également connu sous le nom de Best Effort. Il fournit un système non fiable et sans connexion pour la livraison de paquets. UDP ne fournit pas de mécanisme pour garantir la livraison et la commande, laissant plutôt les applications de niveau supérieur s'occuper des données perdues ou en panne. UDP est principalement utilisé pour la communication de diffusion ou les jeux vidéo en réseau.
   - **ICMP :** est un protocole qui permet la transmission de messages de contrôle (messages d'erreur et messages d'information) liés aux routeurs ou aux hôtes.
   - **IP :** Utilisé pour gérer les services de datagrammes entre les hôtes. Il gère l'adressage, le routage, la fragmentation et le réassemblage des paquets.
3. **Adresses IP source et destination**
   Pour chaque règle de SNORT, il faut une adresse IP source et destination qui permettent de définir la source et la distination du flux de paquets à surveiller. [12]
   Et voici tous les choix possible :
   - Les variables HOME-NET, EXTERNAL-NET en tant que variables (par exemple ipvar HOME-NET 10.136.3.0).
   - mettre en dur dans le fichier les adresses (par exemple : alerttcp 192.168.0.5 any ->...).
   - mettre any (pour toutes les adresses).
   - gérer l'adressage avec des négations grâce à l'opérateur "!" (par exemple alert tcp !19.168.0.5 any ->...).
   - créer une liste variable dans le fichier "snort.conf" que l'on va utiliser directement dans la règle (par exemple ipvar LST-IP [10.136.1.5, 10.136.1.6]).
4. **Port source et destination**
   Les ports permetent d'optimiser les règles pour ne pas passer par tous les ports. On a plusieurs possibilités : [12]
   - mettre any comme port pour prendre en compte tous les ports

- définir statiquement un port (par exemple 80)
- définir une liste de ports (par exemple portvar PORT-HTTPS [36,80])
- mettre un port avec une négation (par exemple !80).

5. **Opération**
   Les opérateurs de directions vont indiquer le sens du trafic à observer. C'est important car, le but n'est pas d'analyser tout le trafic entrant ou sortant. [13]
   Voici les possibilités d'opérateur :
   - Dans un seul sens, par exemple -> ou alors <-
   - Dans les deux sens donc bidirectionnel : <>

2.2.2 Partie option

Les options des règles présentent le cœur du moteur de détection d'intrusion de SNORT, combinant facilité d'utilisation, puissance et flexibilité. Toutes les options sont séparées les unes des autres par un caractère point-virgule "; ". Les mots clés des options sont séparés de leurs arguments avec un caractère deux points " : ".

Il existe plusieurs options de règle disponibles dans SNORT, voici quelques options : [14]

- **Msg (message) :** affiche un message dans les alertes et journalise les paquets.
  Format : msg : "<message texte>";
- **Logto :** l'option va indiquer à Snort d'enregistrer les logs de tous les paquets après une règle spécifique, dans un nouveau dossier log.
  Format : logto : "<nom de fichier>";
- **TTL :** L'option va vérifier la durée de vie de paquets. Elle est utilisée pour détecter les tentatives de trace route.
  Format ttl : "<nombre>";
- **TOS :** Le mot clé "tos" vous permet de vérifier de champ TOS de l'entête IP pour une valeur spécifique. Le test effectué est réussi seulement sur une correspondance exacte.
  Format tos : "<nombre>";
- **ID :** L'option est faite pour vérifier l'ID du paquet IP.
  Format id : "<nombre>";
- **Content (contenu) :** C'est une option très importante. Elle va permettre à l'utilisateur de chercher à l'intérieur d'un PayLoad le contenu qu'il souhaite.
  Format : (content : "POST";)
- **Offset :** L'option est comme depth mais la recherche se fait à partir du nombre indiqué.
  Format : offset : <nombre> ;
- **Flags (Drapeaux) :** L'option est utilisée pour vérifier si des flags TCP sont présents.
  Format : <valeurs de drapeaux> ;
- **Seq (Séquence TCP) :** Cette option se réfère aux numéros de séquence TCP. Elle détecte si le paquet a un numéro de séquence statique fixé, et donc plutôt peu utilisé. Elle a été incluse pour assurer l'exhaustivité.
  Format : seq : <nomber> ;
- **ipoption :** Regarde les champs des options IP pour des codes spécifiques. Format : ipopts :<option> ;

2.3 Fonctionnement de SNORT

SNORT analyse toutes les données et les paquets reçus sur le réseau, puis effectue une action lorsque ces données ne sont pas conformes à des règles bien définies. SNORT peut être configuré pour fonctionner en trois modes : le mode sniffer, le mode packetlogger et le mode détection d'intrusion.

2.3.1 Mode "sniffer"

SNORT lit les paquets circulants sur le réseau et les affiche en permanence sur la console. Il permet d'observer en temps réel le trafic sur le réseau. [13]

2.3.2 Mode "packet logger"

Il enregistre le trafic réseau dans un répertoire sur le disque ou dans une base de données. Ainsi, on n'est pas obligé de rester devant l'écran, mais il suffit juste de consulter les fichiers d'enregistrement de trafic actif sur le réseau. [13]

2.3.3 Mode "détection d'intrusion sur le réseau"

Dans ce mode, SNORT analyse le trafic réseau, il applique des règles sur tous les paquets capturés. Si un paquet correspond à une règle, alors une alerte est générée. Sinon, le paquet est abandonné et aucune entrée de journal n'est créée. Pour démarrer SNORT en mode de détection d'intrusion réseau, utilise la commande : snort -c /opt/snort/etc/snort.conf. [7]

2.4 Composants de SNORT

SNORT comporte plusieurs composants. Ces composants fonctionnent ensemble pour détecter des attaques particulières et générer une sortie dans un format requis. Un IDS basé sur SNORT comprend les principaux composants suivants :

- Décodeur de paquets
- Préprocesseurs
- Moteur de détection
- Système de journalisation et d'alerte
- Module de sortie

La figure 2.2. montre les composants de SNORT. [7]

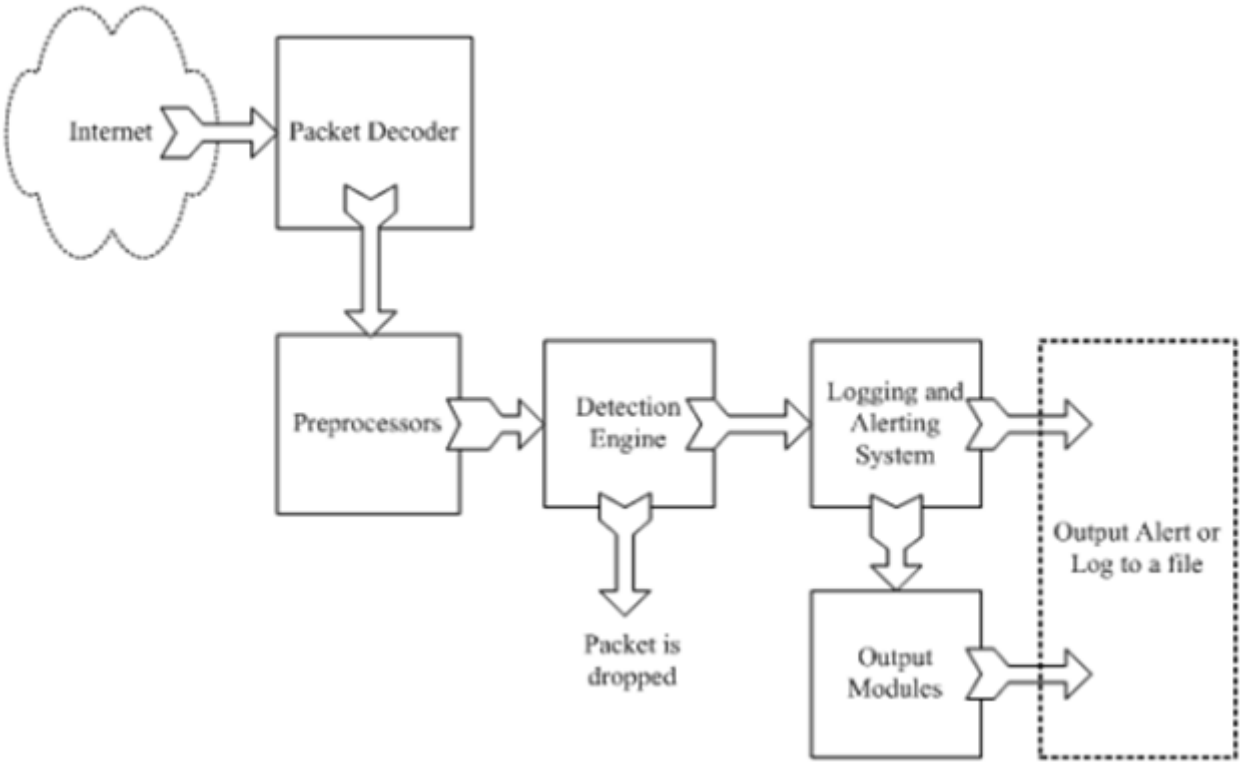

FIGURE 2.2 – Composants du SNORT.

Nous allons détailler chaque composant dans la section suivante.

2.4.1 Décodeur de paquets

Système de détection d'intrusion lire et analyser tous les paquets passants par le lien de communication et cela par l'activation d'une ou plusieurs interface en mode espion (promiscuous mode). SNORT utilise la bibliothèque libcap pour faire la capture des trames.
Le décodeur de paquets se compose de plusieurs sous-décodeurs organisés par protocole (ethernet,ip,tcp...), qui convertissent les éléments du protocole en structures de données internes [4], comme illustrée la figure 2.3.

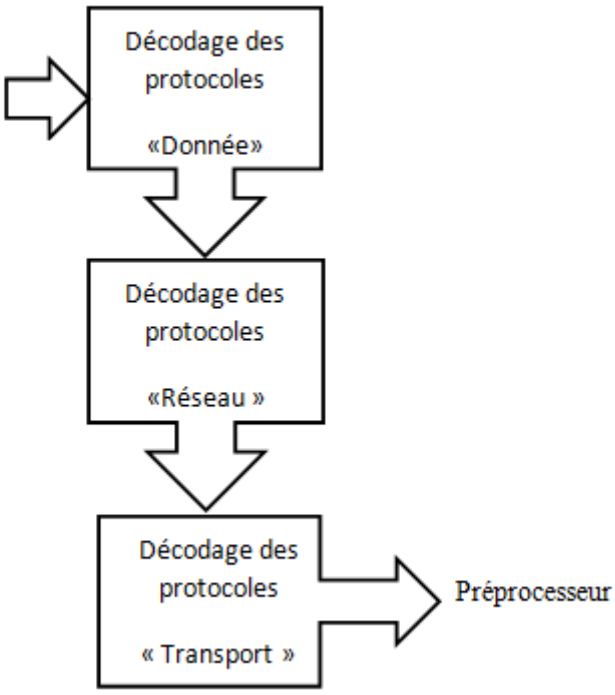

FIGURE 2.3 – Décodeur de paquets.

2.4.2 Préprocesseurs

Les préprocesseurs sont des composants ou des plugins qui peuvent être utilisés avec Snort pour programmer ou modifier des paquets avant que le moteur de détection n'effectue une opération pour savoir si les paquets sont utilisés par un intrus. Certains préprocesseurs effectuent également une détection en trouvant des anomalies dans les en-têtes de paquets et en générant des alertes. Les préprocesseurs sont très importants pour tout IDS, car ils préparent les paquets pour analyse selon les règles du moteur de détection. [7]

2.4.3 Moteur de détection

C'est la partie la plus importante d'IDS. Son travail consiste à détecter la présence d'intrusions dans les paquets, en utilisant les règles Snort. ces derniers sont lues dans une structure de données interne ou une chaîne où elles sont comparées à tous les paquets. Si le paquet correspond à la règle, les mesures appropriées seront prises, sinon le paquet sera supprimé. Une action appropriée peut être de consigner le package ou de générer une alerte. [7]
Ces régle doivent être activés dans le fichier de configuration snort.conf dont une partie est illustrée dans la figures 2.4.

```
include $RULE_PATH/local.rules
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
```

FIGURE 2.4 – Une partie du fichier snort.conf.

2.4.4 Système de journalisatione et d'alerte

Les paquets peuvent être utilisés pour consigner l'activité ou générer des alertes en fonction de ce que le moteur d'inspection trouve dans les paquets. Les journaux sont enregistrés dans des fichiers texte brut, des fichiers tcpdump ou autres. Par défaut, tous les fichiers journaux sont stockés dans le dossier /var/log/snort. [7]
La figure 2.5 présente une maquette qui permettra de tester SNORT afin de mettre en avant ses fonctions. [15]

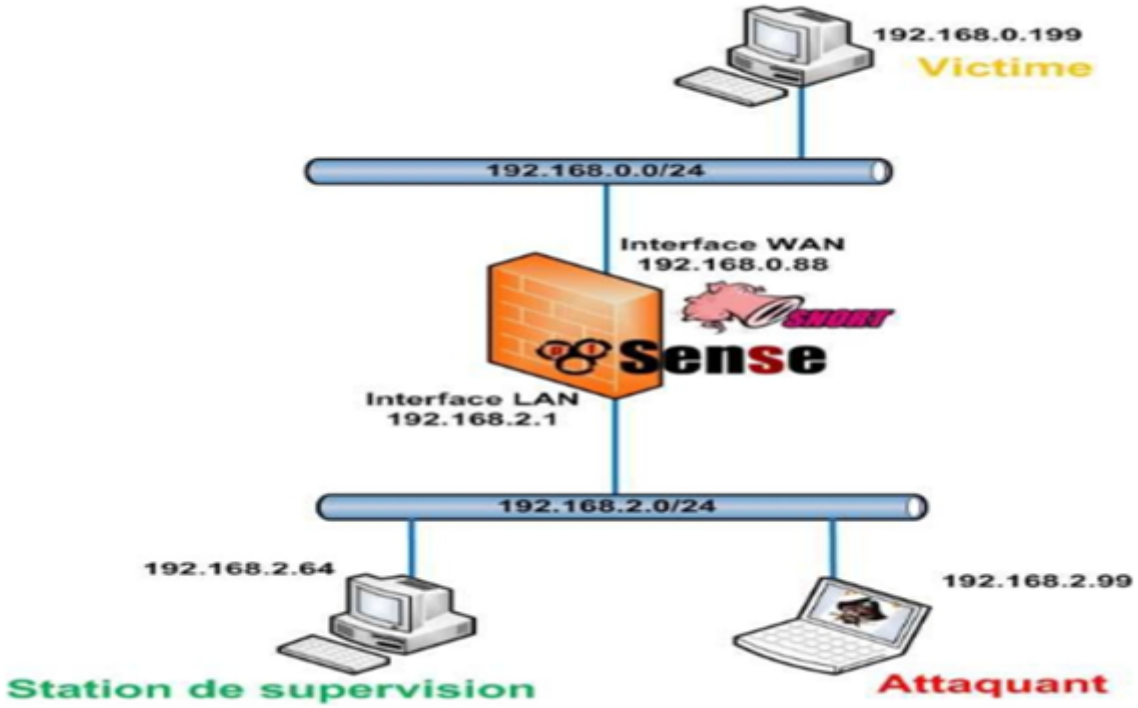

FIGURE 2.5 – Maquette de test des fonctions de SNORT.

2.4.5 Module de sortie

Les modules de sortie ou les plugins peuvent faire différentes choses, selon la façon dont vous souhaitez que la sortie générée par le système de journalisation et d'alerte de Snort soit enregistrée. En fait, ces modules contrôlent le type de sortie générée par le système de journalisation et d'alerte.
Selon la configuration, le module de sortie peut effectuer les actions suivantes :
– Se connecter simplement au fichier /var/log/snort/alerts ou à un autre fichier.
– Envoi d'interruptions SNMP.
– Envoi de messages à la fonction syslog.
– Connexion à une base de données comme MySQL ou Oracle.
– Génération d'une sortie XML (eXtensible Markup Language).
– Modification de la configuration sur les routeurs et les pare-feux.
– Envoi de messages SMB (Server Message Block) à des machines basées sur Microsoft Windows.
D'autres outils peuvent également être utilisés pour envoyer des alertes dans d'autres formats tels que des messages électroniques ou la visualisation d'alertes à l'aide d'une interface Web. [7]

2.5 Outils de SNORT

2.5.1 SNORT raport :

– S'applique à MYSQL et POSTGRESQL.
– Représentation graphique à l'aide de la bibliothèque Jpgraph.
– Visualiser un graphique circulaire en snort (TCP, UDP, ICMP, Portscan).
– Afficher les dernières alertes (intervalle de temps) ou les alertes hebdomadaires (quotidienne).
– Les rapports indiquent des liens vers des sites Web pour exprimer des alertes.

2.5.2 SNORT-Rep

Est un outil à rapporter le SNORT par deux formats (texte et HTML).
Chaque rapport contient un résumé sur :
– Le balayage de port (port-scan).
– Les alertes par ID.
– Les alertes par host éloigné et ID.
– Les alertes par host local et ID.
– Les alertes par port local et ID.

2.5.3 Acid

ACID est un moteur d'analyse PHP de base pour la recherche et le traitement de bases de données d'événements de sécurité générés par des logiciels de sécurité tels qu'un IDS. Cependant, il peut être considérer ACID comme outil d'exportation de rapports, car il affiche des informations statiques d'alerte telles que des graphiques et des tableaux. [12]

2.6 Positionnement de SNORT

L'emplacement physique de la sonde SNORT sur le réseau a un impact considérable sur son efficacité. Dans le cas d'une architecture classique, composée d'un Firewall et d'une DMZ, trois positions sont généralement envisageables : avant le firewall, sur le DMZ, ou sur le réseau interne.

2.6.1 Avant le Firewall ou le routeur filtrant

Dans cette position, le SNORT est à la pointe de la détection des attaques contre l'entreprise provenant de sources externes. SNORT pourra alors analyser tout trafic bloqué par le pare-feu. [16]
Cette position possède deux inconvénients :
– Le risque créé par un trafic très important peut impliquer une perte de fiabilité.
– Être en dehors de la protection du pare-feu et exposer ainsi le NIDS à d'éventuelles attaques et le rendre inefficace.
La figure 2.6 présente l'emplacement de SNORT avant le firewall. [8]

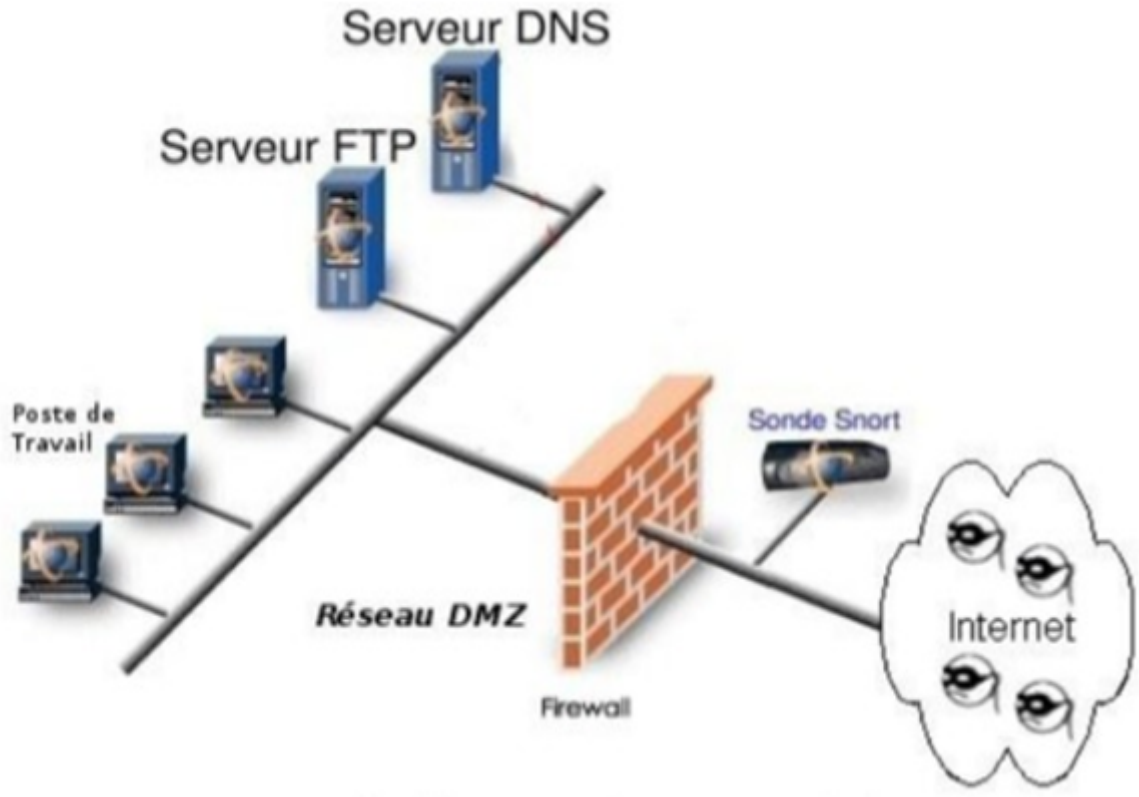

FIGURE 2.6 – Positionnement de snort avant le frewall.

2.6.2 Sur la DMZ

Dans cette architecture, SNORT peut surveiller les attaques contre les différents serveurs de l'entreprise accessibles de l'extérieur, et aussi peut détecter tout le trafic filtré par le pare-feu et atteignant la zone DMZ. [16]
La figure 2.7 présente l'emplacement de SNORT sur la DMZ. [8]

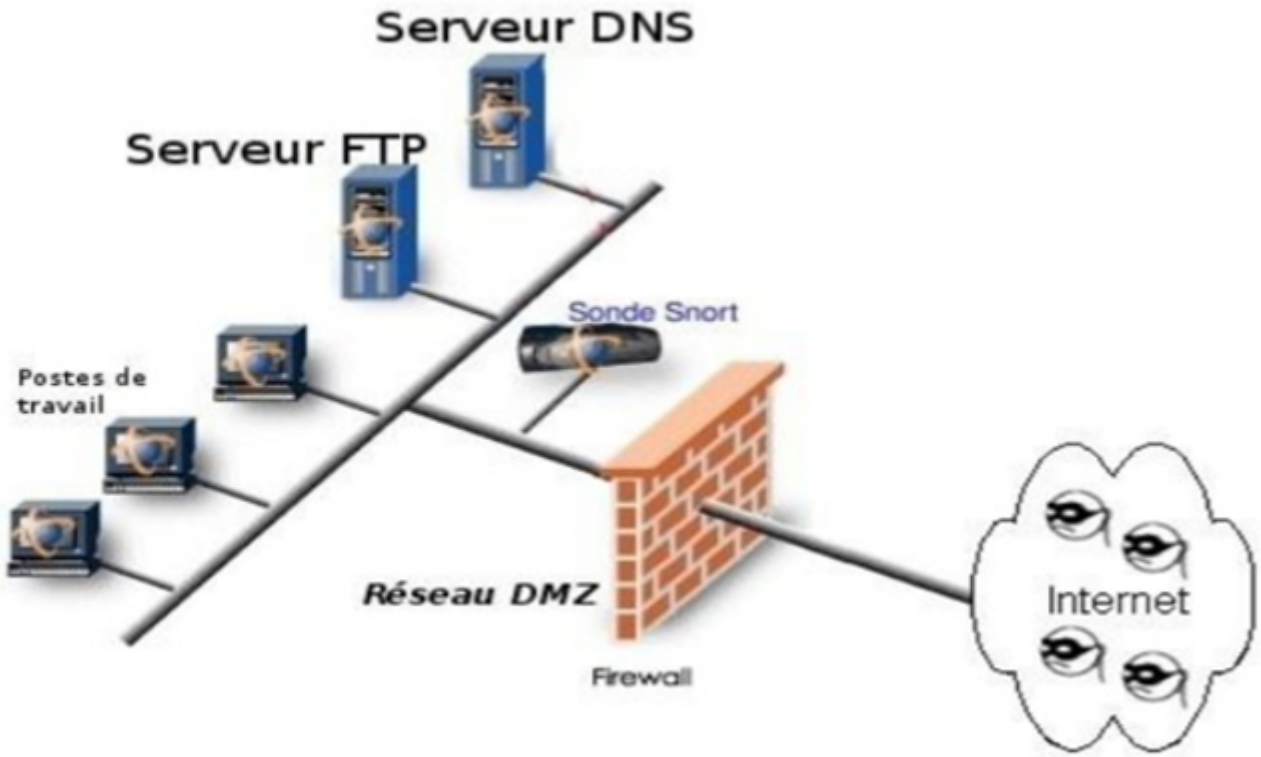

FIGURE 2.7 – Positionnement de snort sur DMZ.

2.6.3 Sur le réseau interne

Placer SNORT a cet endroit permet de surveiller les tentatives d'intrusion depuis le réseau d'entreprise et les tentatives d'attaque depuis l'intérieur. Ce poste est probablement le plus important si votre entreprise utilise principalement des outils informatiques pour gérer ses activités et ses réseaux, car il est accessible à des personnes moins soucieuses de la sécurité (réseaux scolaires ou universitaires). [16]
La figure 2.8 présente l'emplacement de SNORT sur le réseau interne. [8]

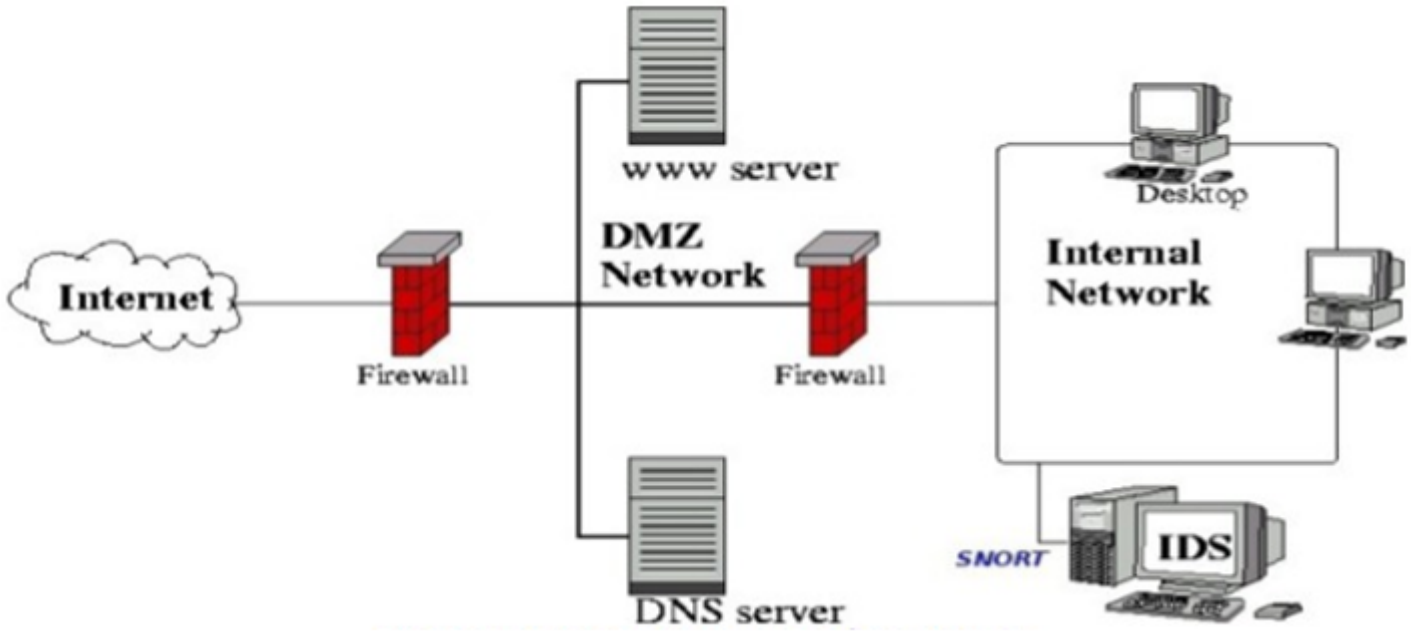

FIGURE 2.8 – Positionnement de snort sur le réseau internet.

2.7 Points forts et points faibles de SNORT

SNORT présente des avantages et des inconvenients

2.7.1 Points forts de SNORT

- C'est un logiciel open source et gratuit.
- Il est disponible sur la plupart des systèmes d'exploitation (Windows et linux comme Ubuntu, Debian, CentOS).
- Les mises à jour des règles sont gratuites.
- Il est capable d'effectuer une analyse en temps réel et du trafic entrant et sortant.
- La détection et la notification des attaques sont déjà connues.

2.7.2 Points faibles de SNORT

- Les faux positifs ou alertes, lorsque SNORT donne un avertissement alors qu'il ne devrait pas.

Conclusion

Le seul inconvénient est que les systèmes ne peuvent jamais être totalement sécurisés. L'objectif est donc de pouvoir détecter autant d'intrusions que possible. SNORT est capable de détecter et de bloquer plusieurs intrusions simultanément grâce à ses règles. C'est un véritable portail anti-intrusion.
Nous avons présenté dans ce chapitre les notions de base de SNORT, avec ces modes de fonctionnement ainsi que ces règles qui lui permettant d'examiner les attaques malveillantes.
Dans le chapitre suivant, nous étudierons l'organisme d'acceuil CEVITAL, l'une des entreprises les plus célèbres non seulement en Algérie mais aussi dans le monde entier.

Chapitre 3

Organisme d'accueil

Introduction

Cevital est la première entreprise privée algérienne à avoir investi dans des secteurs d'activités diversifiés. Elle a traversé d'importantes étapes historiques pour atteindre sa taille et sa notoriété actuelle.
Afin de compléter notre travail théorique des deux chapitres précédents, nous allons étudier et analyser l'environnement existant du groupe CEVITAL.
Notre stage est effectué au niveau du département Réseau de Télécom de la direction de système d'information (DSI).
Dans ce chapitre, nous allons présenter l'organisme d'accueil, en citant les différents départements qui le constituent, avec sa situation géographique. Ensuite, définir la problématique.

3.1 Présentation de l'entreprise

3.1.1 Cevital agro-industrie

Créée en 1998, CEVITAL Agro-industrie est le leader du secteur agroalimentaire en Algérie. Implantée au sein du port de Bejaia (Algérie), CEVITAL Agro-industrie est composée de plusieurs unités de production telles que : la raffinerie d'huile, raffinerie de sucre, margarinerie, unité de conditionnement d'eau minérale, unité de fabrication et de conditionnement de boisson rafraichissante, conserverie, silos portuaires ainsi qu'un terminal de déchargement portuaire.
CEVITAL Agro-industrie offre des produits de qualité supérieure à des prix compétitifs. Grâce à son savoir-faire, ses unités de production ultramodernes, son contrôle strict de qualité et son réseau de distribution. Elle couvre les besoins nationaux et a permis de faire passer l'Algérie du stade d'importateur à celui d'exportateur pour les huiles, les margarines et le sucre. [17]

3.1.2 Cevital géographique

CEVITAL est l'une des plus grandes entreprises en Algérie et un leader dans le secteur agroalimentaire. Sa base de production est située au nouveau terminal du port de Béjaïa, à 3 km au sud-ouest de la ville, à proximité de la route nationale 26. Cette localisation de l'entreprise bénéficie de sa proximité économique. En fait, il est très proche du port et de l'aéroport. Le complexe couvre une superficie de 45 000 mètres carrés (le plus grand complexe privé d'Algérie). Il dispose d'une capacité de stockage de 182 000 tonnes/an (silos portuaires) et d'un terminal portuaire de déchargement (réception des matières premières) de 200 000 tonnes/heure.
La figure 3.1 illustre un vu satellitaire du complexe de CEVITAL.

FIGURE 3.1 – Vue satellitaire du complexe CEVITAL.

3.1.3 Historique

ISSAD Rebrab fonde son cabinet d'experts-comptables en 1968, puis se lance en 1971 en créant des sociétés dans l'industrie métallurgique et en 1991 dans la sidérurgie. A la tête du Groupe CEVITAL, il poursuit son développement à travers une diversité d'activités et compte aujourd'hui 26 filiales dans les secteurs de l'industrie, de l'agroalimentaire, de la grande distribution et de l'automobile. Elle opère depuis plusieurs années à l'international, notamment en Europe (France, Italie, Espagne), en Tunisie, au Maroc et au Brésil. En France, IssadRebrab a racheté la PME Oxxo (fabricant de fenêtres performantes) en 2013, Brandt France en 2014 pour relancer le leader de l'électroménager, et Luccini, une entreprise en 2015. Complexe sidérurgique spécial. Issad REBRAB est Président du Conseil d'Administration du Groupe CEVITAL depuis 2008. Père de 5 enfants, tous impliqués dans la gestion du groupe. En 2009, CEVITAL a élargi sa gouvernance en décidant d'ouvrir son conseil d'administration à des membres indépendants. Des décisions prises dans une volonté d'accompagner la croissance de l'entreprise et d'assurer sa pérennité, à l'image d'une grande entreprise internationale. Après l'acquisition de l'activité sidérurgique de Piombino, IssadRebrab a été sélectionné comme PDG de l'année lors de l'Africa CEO Forum 2015 et a été nommé Person of the Year par la région Toscane (Italie) en 2016. [18]

3.1.4 Activités de cevital

Les activités du CEVITAL agro-industrie au niveau de l'Algérie sont répartit comme suit : [17]

a) Activité de CEVITAL au niveau de la commune Bejaia

Au niveau de la commune de Bejaia, l'entreprise CEVITAL fait la contribution des installations suivantes (l'industrie agro-alimentaire) :

- La production de la margarinerie.
- Le raffinage de sucre.
- Le raffinage des huiles alimentaires.
- Silos portuaires

b) Activité de CEVITAL au niveau de la commune d'EL kseur

Au niveau de la commune d'El-kseur (Béjaia), on trouve l'unité de production du jus de fruits COJECK qui a été racheté par le groupe CEVITAL dans le cadre de la préservation des entreprises publiques algériennes en novembre 2006.

c) Activité de CEVITAL au niveau de la commune Agouni Gueghane

Au niveau de la commune de AgouniGueghane (Tizi ouzou) dans les montagnes de Djurdjura, le groupe CEVITAL a inauguré en 2007, l'unité de production d'eau minérale Lalla khedidja.

3.1.5 Infrastructure de l'entreprise

Les infrastructure du groupe CEVITAL sont comme ce suit :

- Deux (02) raffineries de sucre.
- Une unité de sucre liquide.
- Une raffinerie d'huile.
- Une margarinerie.
- Une unité de conditionnement d'eau minérale (site de tiziouzou).
- Une unité de fabrication et de conditionnement de boissons rafraichissantes (site d'EL Kseur).
- Une conserverie.
- Une unité de fabrication de chaux calcinée.

3.2 Organigramme générale de CEVITAL

Le groupe CEVITAL est composé de huit (08) directions principales dont chacune a pour but d'assurer le bon fonctionnement de ces tâches comme le montre l'organigramme de la figure 3.2 :

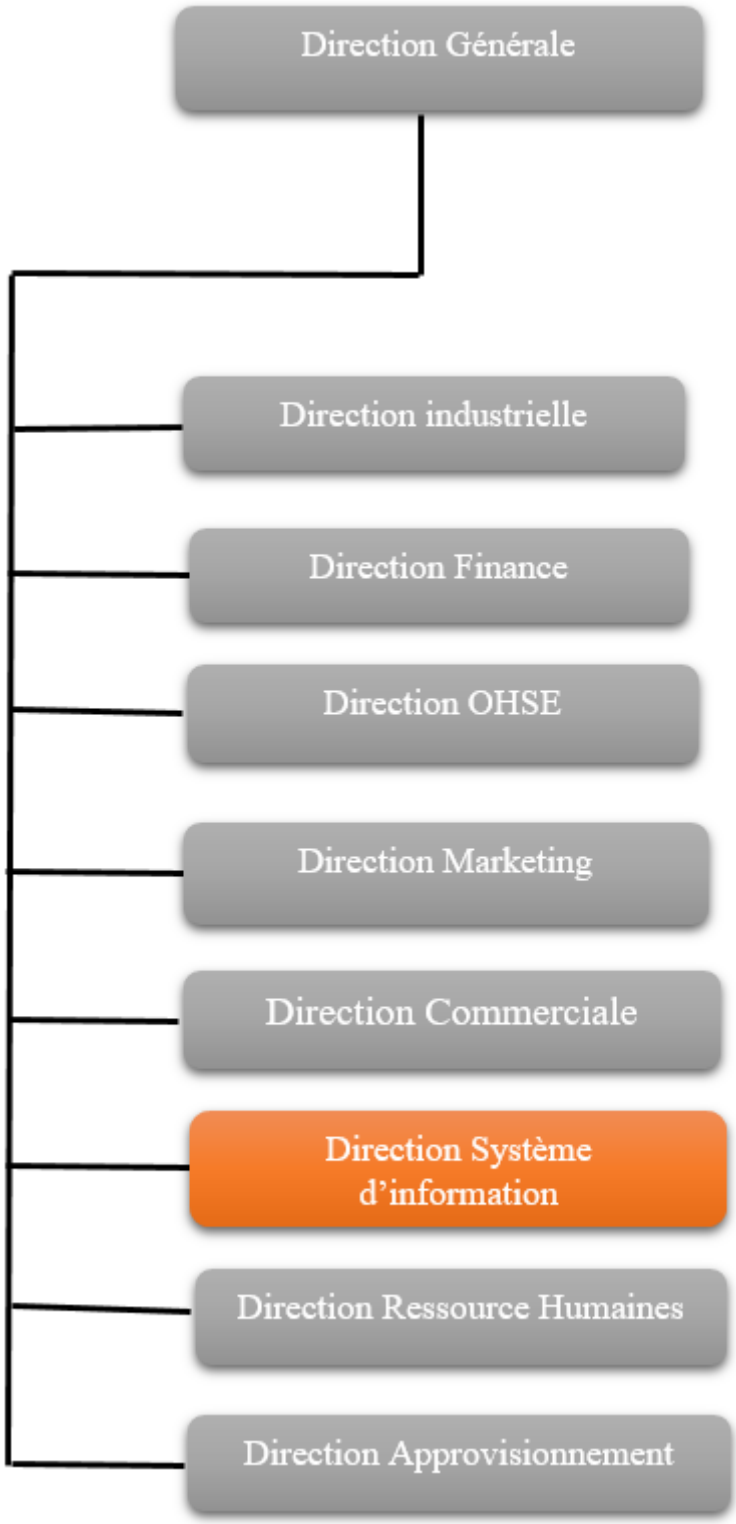

FIGURE 3.2 – Organigramme du groupe CEVITAL.

3.3 Audit du réseau informatique de CEVITALE

3.3.1 Définition Audit

L'audit informatique est une démarche qui consiste à affirmer que les activités informatiques d'une entreprise ou d'une direction sont menées conformément aux règles et pratiques professionnelles. [19]

3.3.2 Champs d'étude

Nous effectuons un stage dans la section "Réseaux Télécom" de la Direction des Systèmes d'Information (DSI). Ce dernier assure la mise en place des outils informatiques nécessaires pour soutenir et améliorer les activités, les stratégies et la performance de l'entreprise.
Il doit donc assurer la cohérence, l'évolution, la maîtrise technique et la mise à disposition en toute sécurité des moyens informatiques et de communication mis à la disposition des utilisateurs. Il identifie également les évolutions nécessaires en fonction des objectifs de l'entreprise et des nouvelles technologies dans le cadre d'un plan pluriannuel.
La figure 3.3. montre l'organigramme de direction des systèmes d'informations (DSI).

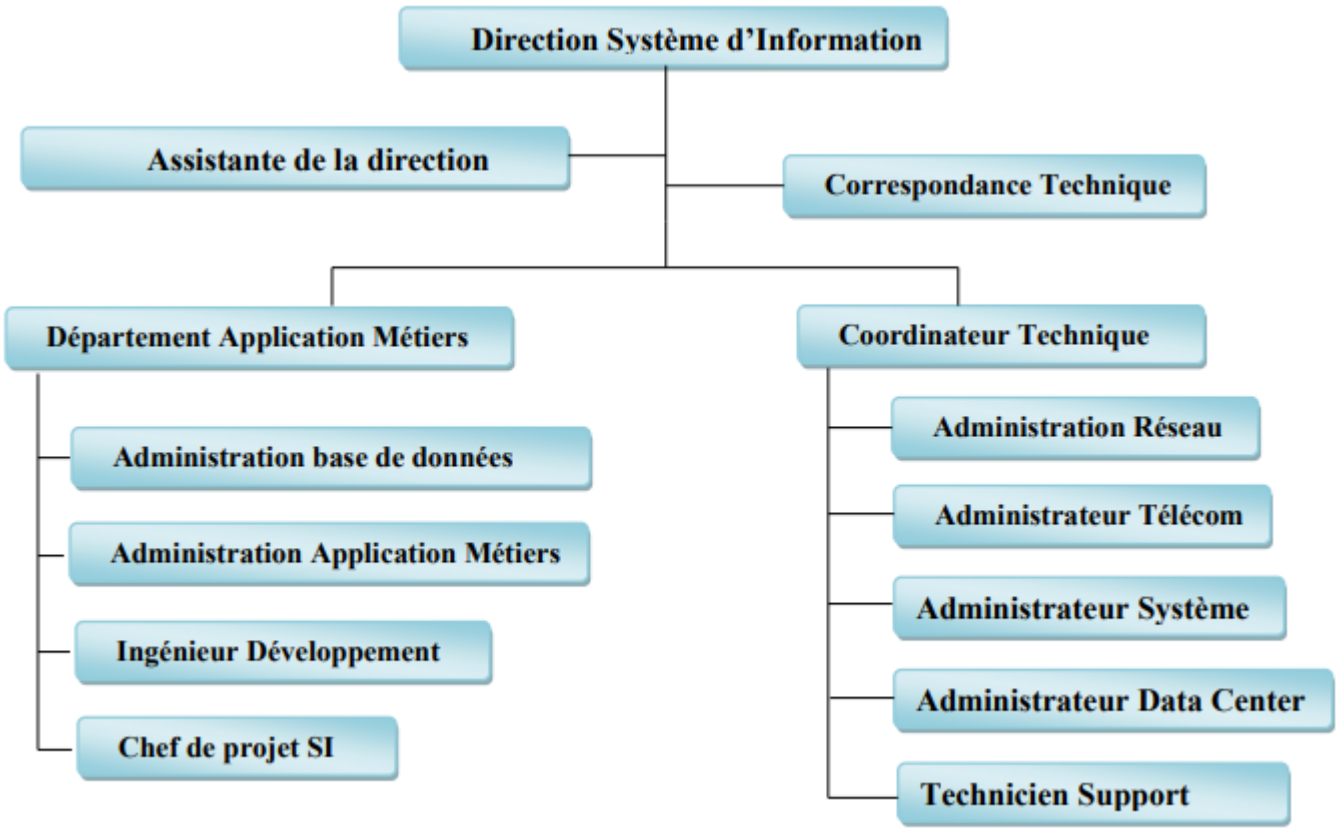

FIGURE 3.3 – Organigramme de DSI.

3.4 Etude de l'existant

3.4.1 Phase d'analyse

Après la visite de la salle des équipements (data center) on a remarqué que l'entreprise utilise beaucoup plus les équipements de CISCO concernant leur réseau informatique ainsi que PALOAlTO (pour les firewall) et ALCATEL pour la téléphonie.
La sauvegarde de la configuration des équipements réseau (routeurs, switchs, pare-feu...) se fait manuellement par l'invite de commande CLI.

3.4.2 Sécurité

La politique de sécurité de CEVITAL repose sur deux parties : partie logique et partie physique.

a) Sécurité logique

Consiste à protéger le réseau contre les attaques de piratage et l'écoute clandestine. Elle se fait à l'aide de :

1. **Pare-feu** Le pare-feu utilisé pour cette partie est un paloalto 3020 connecté à un commutateur de distribution Catalyst 4705R et à la DMZ de l'autre côté. Il est doté de fonctions unifiées de gestion des menaces : antivirus, antispam, système de prévention des intrusions et filtrage des contenus web. Les administrateurs réseau utilisent une politique de sécurité pour contrôler le trafic via ce pare-feu.
2. **Serveur Proxy** C'est un serveur Microsoft qui se situe entre le pare-feu et le commutateur principal. Sa mission est de fournir une gestion simplifiée et c'est une passerelle Web sécurisée qui protège les employés lors de leur navigation sur Internet.
3. **DMZ :(zone démilitarisée)** Une DMZ est mise en place pour permettre aux clients de se connecter au réseau à partir de sites externes. Il se compose d'un serveur FTP relié au pare-feu. Son rôle est d'identifier ceux qui souhaitent y accéder afin d'en restreindre l'accès en laissant un chemin sécurisé vers la base de données CEVITAL.

b) Sécurité physique

1. **Accès au local des équipements réseau :** L'accès local aux équipements de technologie de réseau géré est protégé par un système de contrôle d'accès et accessible uniquement aux :
   - Instalateur, Agent de Maintenance.
   - Administrateur réseau.
   - Les personnes chargées de la sécurité du data center.
2. **Détecteur d'incendie :** Pour éviter les dommages, un système de protection contre les incendies a été mis en place indépendamment de tous les autres systèmes.
3. **Régulateur de température et faux planchers :** Le matériel et les composants électriques sont extrêmement sensibles à l'humidité. Pour cette raison, les ingénieurs de CEVITAL ont installé des régulateurs de température pour maintenir la température ambiante et des planchers surélevés pour prévenir les dégâts (inondations, ruptures de canalisations, etc.).
4. **Onduleur :** Pour éviter les coupures de courant inattendues, tous les dispositifs du système d'information CEVITAL sont connectés à un circuit ondulé avec une disponibilité supérieure à 1/4 d'heure pour permettre une extension spécifique au système.

3.5 Expression des besoins de sécurité

3.5.1 Critique

Les administrateurs réseaux du Groupe CEVITAL font confiance aux pare-feux pour protéger et sécuriser les réseaux d'entreprise des attaques (virus, hackers, intrus, etc.). Il s'agit d'un problème important car les pare-feu sont 100/100 inefficaces et les réseaux protégés par un pare-feu sont toujours vulnérables aux attaques. Si l'IDS est intégré au pare-feu, le pare-feu entrera en concurrence lors de l'analyse du même flux de données, ce qui peut entraîner le blocage de l'appareil et réduire considérablement les performances en cas de trafic important.
De ce fait, lors de mon stage chez CEVITAL, Nous avons remarqué une anomalie liée à la sécurité du réseau, l'absence de mécanisme pour détecter et prévenir les intrus.

3.5.2 Solution proposée

Afin remédier au problème traité dans la problématique, nous avons choisi de mettre en place un IDS (SNORT) sous le système d'exploitation LINUX qui sera placé avant le pare-feu.
La configuration IDS (SNORT) fonctionne indépendamment des autres appareils, ce qui peut augmenter le niveau de sécurité et les capacités de détection. Il joue le rôle d'un complément aux pare-feu Il le fait en lui permettant d'analyser plus Renseignements sur le trafic.

Conclusion

CEVITAL agro-industrie est le leader du secteur agro-alimentaire en Algérie. Sa mission principal est le développement de sa production afin d'assurer la qualité et les conditions de ses différents produits (huile, margarine, sucre, eau minérale, boissons fruitées) et de satisfaire ses différents clients par la couverture du marché national. Nous avons pu constater que l'entreprise CEVITAL se caractérise par des moyens efficaces (Capital) et outils modernes tel que leur logiciels, et ainsi d'un bon savoir faire, qui l'aide a mieux gérer ses différentes fonctions.
Dans ce chapitre, nous avons présenté l'organisme d'accueil au sein du l'entreprise CEVITAL.
Le chapitre suivant sera consacré à la mise en place de SNORT au sein de cette entreprise.

Chapitre 4

Mise en place de SNORT

Introduction

Dans ce dernier chapitre, nous allons voir un cas pratique concernant snort. Nous allons voir comment installer les différents composants du NIDS ainsi que toutes les configurations nécessaires.
Au final, nous allons tester notre configuration en lançant quelques attaques et essayer de les détecter.
Dans ce suit, nous allons commencer par donner la position choisit pour l'emplacement de SNORT sur un réseau, en suit nous allons présenter leur manipulation : installation, configuration et fonctionnalités.
Enfin, nous allons terminer par donner une conclusion pour ce travail.

4.1 Choix l'architecture

Nous avons présenté les trois emplacements physiques de SNORT sur le réseau qui sont : SNORT avant le Firewall ou le routeur filtrant, SNORT sur DMZ, SNORT sur le réseau interne. Vue que l'entreprise CEVITAL s'interesse aux menaces externes provenant du réseau internet, nous avons opté pour l'emplacement SNORT avant le firewall ou le routeur filtrant à cause de ses avantages sur la sécurité contre ces menaces. La figure 4.1. montre emplacement de SNORT choisie.

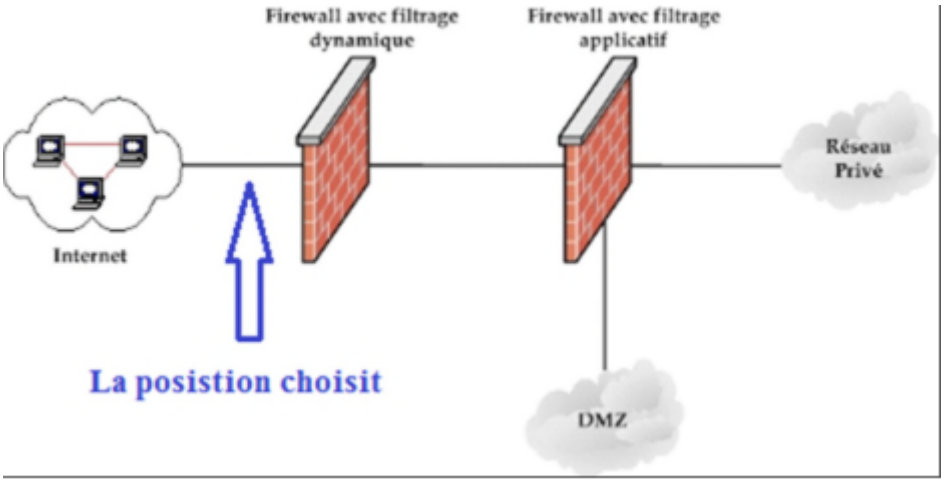

FIGURE 4.1 – Emplacement de SNORT choisie.

4.2 Environnement

Nous avons préféré de travailler dans un environnement Ubuntu, car il nous fournit un espace de travail unique et nous assure une fiabilité de résultats. Ce dernies est s'installer sur VirtualBox.

- **VirtualBox** est un logiciel virtualisation gratuit, open source et multiplateforme d'Oracle. Cela nous permet d'héberger une ou plusieurs machines virtuelles avec différents systèmes d'exploitation. Le logiciel fonctionne sur différents systèmes d'exploitation invités Windows, Linux, MacOS et Solaris. Avec ces systèmes d'exploitation invités, nous pouvons tester des logiciels sur une machine virtuelle sans risque d'endommager l'ordinateur (hôte). [20]
- **Ubuntu** est un système d'exploitation logiciel open source sur Linux, crée en 2004 par Mark Shuttleworth. [21]

4.3 Mise en place de SNORT

Pour initialiser SNORT sous Ubuntu, nous devons d'abord installer les outils de compilations et les dépendances de Snort :

4.3.1 Compilations de SNORT

- **Libpcap (Packet CAPture) :** Librairie utilisée par Snort pour capturer les paquets.
- **Libnet :** C'est une bibliothèque logicielle open source qui facilite la création de packages sur un réseau.
- **GCC :** Est un compilateur sous linux permettant de compiler du C, du C++ , du java... pour compiler les sources de SNORT.
- **libpcre :** Est une librairie de fonctions utilisant la même syntaxe et sémantique que perle 5.
- **Daq :** C'est une bibliothèque qui permet d'acquérir des signaux des paquets et ainsi les convertir en données manipulables par un logiciel. [22]
- **Zlib :** Est une bibliothèque logicielle de compression de données.

4.3.2 Dépendances de SNORT

- **Barnyard2 :** Barnyard2 est un interpréteur open source pour les fichiers de sortie binaires Snort unified2. Son utilisation principale est de permettre à Snort d'écrire sur le disque de manière efficace et de laisser la tâche d'analyser les données binaires dans divers formats à un processus séparé qui ne fera pas manquer à Snort le trafic réseau.
  Barnyard2 est un moyen de stocker et de traiter les sorties binaires de Snort dans une base de données MySQL. [16]
- **B.A.S.E :** BASE (Basic Analysis Security Engine) est une application développée en PHP, qui permet de gérer l'interface graphique de SNORT dans laquelle SNORT stocke ses alertes. Elle est utilisée pour afficher les logs générés par l'IDS et envoyée dans la base de données. [4]
  Pour fonctionner, BASE a besoin d'un certain nombre de dépendance :
  - Un SGBD installé par exemple MySQL.
  - SNORT compilé avec le support de ce SGBD.
  - Un serveur http, par exemple Apache.
  - PHP5 : module PHP.
  - PHP-MySQL : interface PHP/MYSQL.
  - La bibliothèque ADODB (Active Data objet Data Base), destinée à communiquer avec différentes systèmes de gestion de base de données (SGBD) comme MySQL, SQL server, etc
  - PHP-Mail : extension PHP.

4.3.3 Installation et configuration de SNORT

L'installation et la configuration de SNORT sur ubuntu est assez simple mais nécessite quelques étapes.

- D'abord nous avons commençé par mettre à jour le système. Pour cela, nous avons ouvret un terminal puis nous sommes connectés en tant que **root** et nous avons exécuté les commandes suivantes :
  sudo apt-get update
  sudo apt-get upgrade
- Ensuite, nous avons installé tous les logiciels prérequis pour préparer nos serveurs cloud à l'installation de SNORT avec la commande suivante :

sudo apt-get install -y gcc libpcre3-dev zlib1g-dev libluajit-5.1-dev libpcap-dev openssl libssl-dev libnghttp2-dev libdumbnet-dev bison flex libdnet autoconf libtool

– puis nous avons modifié **/etc/network/interfaces** comme un admin :
sudo pico /etc/network/interfaces

– Nous avons rajouté les deux lignes suivantes pour chaque interface réseau :
post-up ethtool -K eth0 gro off
post-up ethtool -K eth0 lro off

– Enfin, nous avons créé un répertoire pour enregistrer les fichiers téléchargé :
mkdir /snort—src
cd /snort—src

a) Installation de SNORT

Pour effectuer les appels abstraits aux bibliothèques de capture de paquets SNORT utilise bibliothèque d'acquisition de données (DAQ).

– Nous avons téléchargé et installé DAQ à partir de site SNORT :
wget https ://www.snort.org/downloads/snort/daq-2.0.7.tar.gz tar -xvzf daq-2.0.7.tar.gz
cd daq-2.0.7
./configure
make
sudo make install

– Une fois daq est installé, nous pouvons démarrer l'installation de SNORT avec les commandes suivantes :
cd /snort—src
wget https ://www.snort.org/downloads/snort/snort-2.9.20.tar.gz
tar -xvzf snort-2.9.20.tar.gz
cd snort-2.9.20
./configure –enable-sourcefire
make
sudo make install

– Pour mettre à jour des bibliothèques partagées, nous avons exécuté la commande suivante :
sudo ldconfig

– Nous avons placer le binaire local de SNORT dans **/usr/local/bin/snort** et créer un lien symbolique vers **/usr/sbin/snort** avec la commande suivante :
sudo ln -s /usr/local/bin/snort /usr/sbin/snort

b) Teste l'installation de SNORT

Pour tester l'installation de SNORT, nous avons utilisé la commande suivante :

/usr/sbin/snort -v

Le résultat illustré dans la figure 4.2 suivante :

FIGURE 4.2 – L'installation de SNORT.

c) Configuration de SNORT

Configuration des règles de SNORT, pour des raison de sécurité, afin que SNORT fonctionne en tant qu'utilisateur non privilégié.

D'abors nous avons commancé par la création de l'utilisateur SNORT et le groupe :

sudo groupadd snort

sudo useradd snort -r -s /sbin/nologin -c SNORT-IDS -g snort

Ensuite nous avons passé a la création la structure de dossiers pour héberger la configuration de SNORT, puis nous avons modifié la propriété de ces fichier pour le nouvel utilisateur SNORT.

Les fichiers de configuration de SNORT sont stocké dans **/etc/snort**, les règles dans **/etc/snort/rules** et **/usr/local/lib/snort—dynamicrules**, et on stock ces journaux dans /var/log/snort.

tous ces étapes avec les commandes suivantes :

- Nous avons tout d'abord créé les repertoire de SNORT :
  sudo mkdir /etc/snort
  sudo mkdir /etc/snort/rules
  sudo mkdir /etc/snort/rules/iplists
  sudo mkdir /etc/snort/preproc—rules
  sudo mkdir /usr/local/lib/snort—dynamicrules
  sudo mkdir /etc/snort/so—rules
- Ensuite nous avons créé les fichiers qui stocke les règles
  sudo touch /etc/snort/rules/iplists/default.blacklist
  sudo touch /etc/snort/rules/iplists/default.whitelist
  sudo touch /etc/snort/rules/local.rules
- Puis nous avons créé les repertoires de journalisation
  sudo mkdir /var/log/snort

sudo mkdir /var/log/snort/archived—logs

- Aussi nous avons réglé les autorisation :
  sudo chmod -R 5775 /etc/snort
  sudo chmod -R 5775 /var/log/snort
  sudo chmod -R 5775 /var/log/snort/archived—logs
  sudo chmod -R 5775 /etc/snort/so—rules
  sudo chmod -R 5775 /usr/local/lib/snort—dynamicrules
- Nous avons changé les propriété sur les dossiers :
  sudo chown -R snort :snort /etc/snort
  sudo chown -R snort :snort /var/log/snort
  sudo chown -R snort :snort /usr/local/lib/snort—dynamicrules
- Nous avons fini par déplacer les fichiers suivants vers /etc/snort avec les commandes suivantes :
  cd /snort—src/snort-2.9.20/etc/
  sudo cp *.conf* /etc/snort
  sudo cp *.map* /etc/snort
  sudo cp *.dtd* /etc/snort
  cd /snort—src/snort-2.9.20/src/dynamic-preprocessors/build/usr/local/lib/snort—dynamicpreprocessor/
  sudo cp * /usr/local/lib/snort—dynamicpreprocessor/
- Le fichier de configuration SNORT est stocké a **/etc/snort/snort.conf**, et contient tous les paramètres que SNORT vas utiliser quand il est éxécuté en mode NIDS.
- Nous avons met en commentaire toutes les règles de Snort avec la commande suivante :
  sudo sed -i 's/include $/RULE/^{-}PATH/include/$**RULE/—PATH/' /etc/snort/snort.conf**
- Nous avons d'abord ouvré le fichier de configuration par la commande :
  sudo pico /etc/snort/snort.conf
- Nous devons indiquer la classe d'adresse du réseau comme suite :
  Dans la ligne 45 on modifie par **ipvar HOME—NET 192.168.109.154/24** comme le montre la figure 4.3.

```
# Setup the network addresses you are protecting
ipvar HOME_NET 192.168.109.154/24
```

FIGURE 4.3 – Classe d'adresse réseau.

- Puis nous avons spécifié le répertoire ou sont disposées nos règles (rules) avec les commandes suivantes :
  Dans la ligne 104 on modifie la ligne par **var RULE—PATH /etc/snort/rules**
  Dans la ligne 105 on modifier la ligne par **var SO—RULE—PATH /etc/snort/rules/so—rules**
  Dans la ligne 106 on modifie la ligne par **var PREPROC—RULE—PATH /etc/snort/preproc—rules**
  Dans la ligne 113 on modifie la ligne par **var WHITE—LIST—PATH /etc/snort/rules/iplists**
  Dans la ligne 114 on modifie la ligne par **var BLACK—LIST—PATH /etc/snort/rules/iplists**
  Comme le montre la figure 4.4

```
var RULE_PATH /etc/snort/rules
var SO_RULE_PATH /etc/snort/so_rules
var PREPROC_RULE_PATH /etc/snort/preproc_rules

# If you are using reputation preprocessor set these
# Currently there is a bug with relative paths, they are relative to where snor$
# not relative to snort.conf like the above variables
# This is completely inconsistent with how other vars work, BUG 89986
# Set the absolute path appropriately
var WHITE_LIST_PATH /etc/snort/rules/iplists
var BLACK_LIST_PATH /etc/snort/rules/iplists
```

FIGURE 4.4 – Disposétion des règles.

– Pour activer les règles qu'on veut utiliser il se fait d'enlevé le "dièse" qui se trouve en début de la ligne, Nous avons édité un seul fichier règle **local.rules** et pour que SNORT l'utilise on doits lui enlevé le "dièse" qui rend la ligne, comme commentaire comme le montre la figute 4.5

```
# site specific rules
include $RULE_PATH/local.rules

#include $RULE_PATH/app-detect.rules
#include $RULE_PATH/attack-responses.rules
#include $RULE_PATH/backdoor.rules
#include $RULE_PATH/bad-traffic.rules
#include $RULE_PATH/blacklist.rules
#include $RULE_PATH/botnet-cnc.rules
#include $RULE_PATH/browser-chrome.rules
```

FIGURE 4.5 – Editer le fichier local.rules.

– Pour écrire sur le fichier **local.rules**, nous avons ouvré d'abord le fichier à l'aide de la commande :
sudo pico /etc/snort/rules/local.rules
– Puis nous avons écris les règles qui nous intéresse le plus :
alert icmp any any –> HOME—NET any (msg :"ICMP test"; sid :10000001 ; rev :001 ;)
– Enfin pour s'assurer le bon fonctionnement de SNORT nous avons exécuté la commande suivante :
sudo snort -T -c /etc/snort/snort.conf
La figure 4.6 illustré le fonctionnement de SNORT.

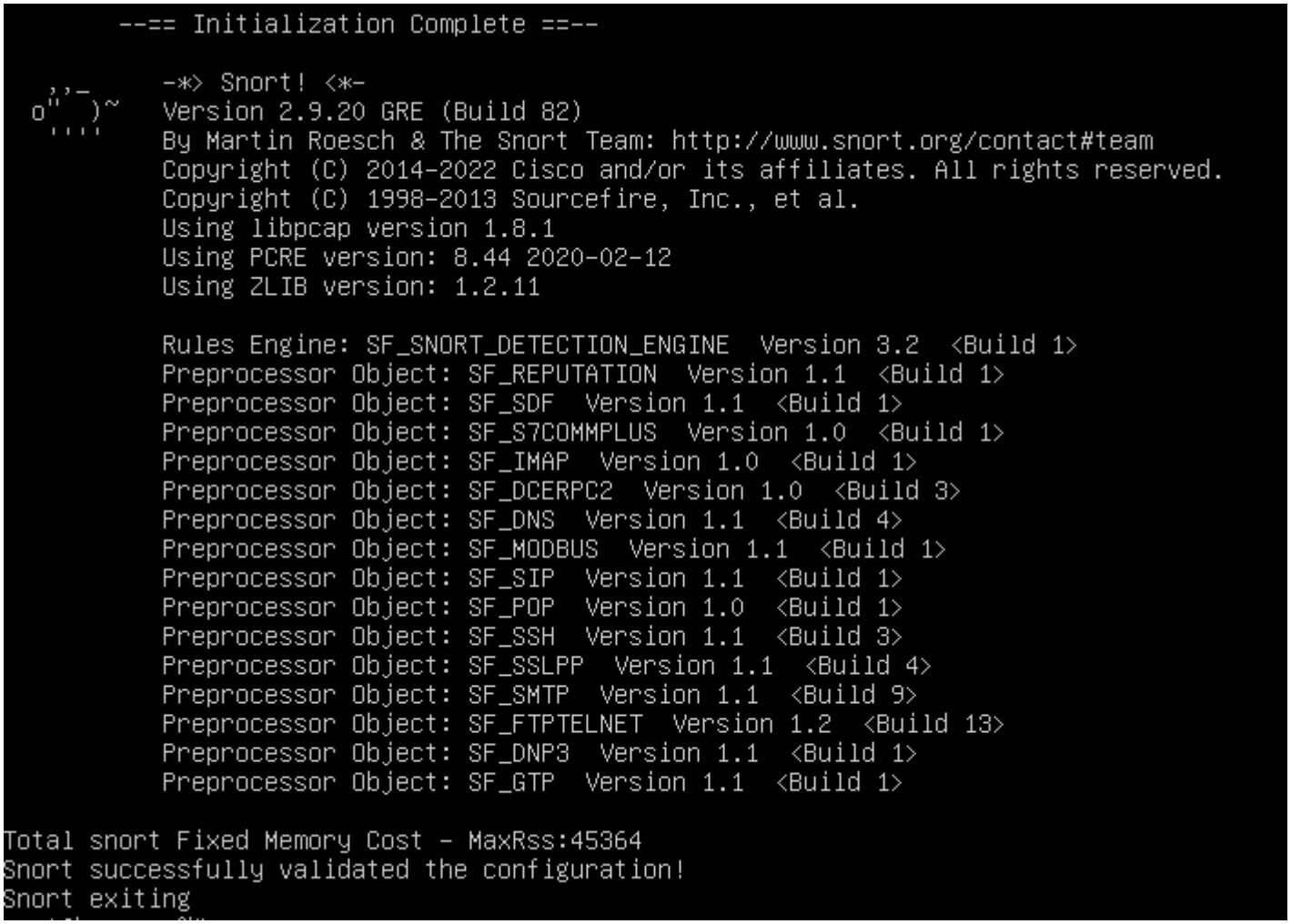

FIGURE 4.6 – SNORT exiting.

4.4 Installation de Barnyard2

4.4.1 Installation des condition préalable

- Tout d'abord, nous avons installé les conditions préalables :
  sudo apt-get install -y mysql-server libmysqlclient-dev mysql-client autoconf libtool
- Ensuite Nous avons modifié le fichier de configuration de SNORT **snort.conf** pour indiqué a SNORT les événements de sortir sous format binaire unified2 pour que Barnyard2 puisse les lire.
- Après la ligne 520 **unified2** nous avons ajouté la ligne suivante :
  output unified2 : filename snort.u2, limit 128, comme le montre la figure 4.7.

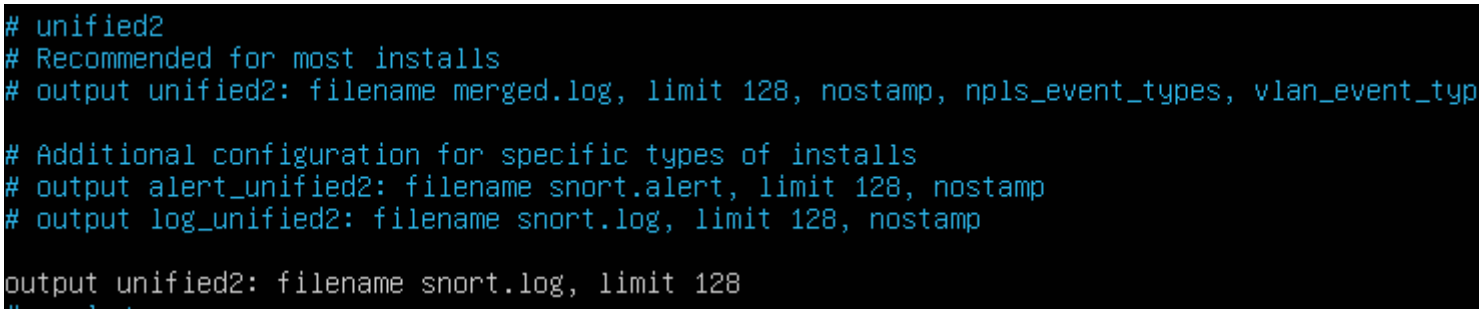

FIGURE 4.7 – Evènements de sortie sous format binaire.

4.4.2 Télécharger, configurer et installer Barnyard2

– Avant tous nous avons accédé au dossier SNORT **/snort—src** avec la commande :
cd /snort—src

– puis nous avons téléchargé Branyard2 dans le site **github.com** avec **wget** :
wget https ://github.com/firnsy/barnyard2/archive/master.tar.gz -O barnyard2-master.tar.gz
La figure 4.8 illustré le téléchargement et l'installation Barnyard2. **tar zxvf barnyard2-master.tar.gz**

FIGURE 4.8 – Téléchargement et installation Barnyard2.

cd barnyard2-master
autoreconf -fvi -l ./m4
– Nous avons placé le binaire local de snort dans **/usr/include/dumbnet.h** et créer un lien symbolique vers **/usr/include/dnet.h** avec la commande suivante :
sudo ln -s /usr/include/dumbnet.h /usr/include/dnet.h
sudo ldconfig

– Selon l'architecture du système (x86 ou x64), exécute une des deux lignes suivantes :
./configure –with-mysql –with-mysql-libraries=/usr/lib/x86—64-linux-gnu
./configure –with-mysql –with-mysql-libraries=/usr/lib/i386-linux-gnu
– Ensuite nous avons passé au commandes suivantes qui permet d'effectue la compilation :
make

sudo make install

4.4.3 Tester l'installation de Barnyard2

Après l'installation, nous avons vérifié si tout est bon en tapant la version du Barnyard2 :
/usr/local/bin/barnyard2 -v
Nous avons eu le résultat comme dans la figure 4.9 :

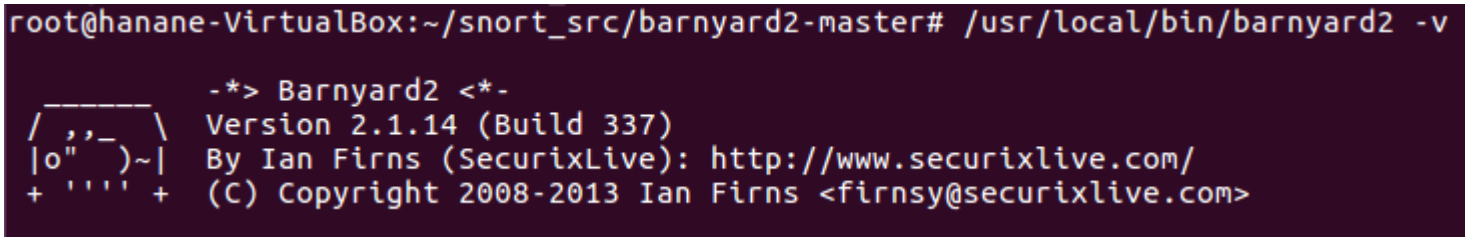

FIGURE 4.9 – Tester l'installation de Barnyard2.

Barnyard2 est installer dans **/usr/local/bin/barnyard2**

4.4.4 Configuration SNORT pour qu'il puisse utiliser Barnyard2

Nous avons besion de quelques fichier pour que SNORT puisse utiliser Barnyard2.
- Tout d'abord nous avons accédé au fichier de snort et Barnyard2 avec la commande :
  cd /snort—src/barnyard2-master
- puis nous avons copié le fichier barnyard2.conf vers le répertoire /etc/snort afin de paramétrer Snort avec barnyard2 :
  sudo cp etc/barnyard2.conf /etc/snort
- Ensuite nous avons créé un dossier ou Barnyard2 stocke les log :
  sudo mkdir /var/log/banyard2
- enfin nous avons réglé les autorisations et changé les propriété sur les dossier comme ce suit :
  sudo chown snort.snort /var/log/barnyard2
  sudo touch /var/log/snort/barnyard2.waldo
  sudo chown snort.snort /var/log/snort/barnyard2.waldo
  sudo touch /etc/snort/sid-msg.map
- Nous avons configurer MYSQL pour stocker les alertes et autres événements générés par snort. Pour cela nous avons se connecte à la base de données MYSQL entant que root :
  mysql -u root -p
- Et nous vous créé la base de donné comme suit :
  CREATE DATABASE snort ;
  USE snort ;
  source /snort—src/barnyard2-master/schemas/CREATE—MYSQL
  CREATE USER 'snort'@'localhost' IDENTIFIED BY 'liza2013' ;
  Grant ALL on snort.* to 'snort'@'localhost' ;
  exit ;
  La figure 4.10 montre la création de base de donnée SNORT

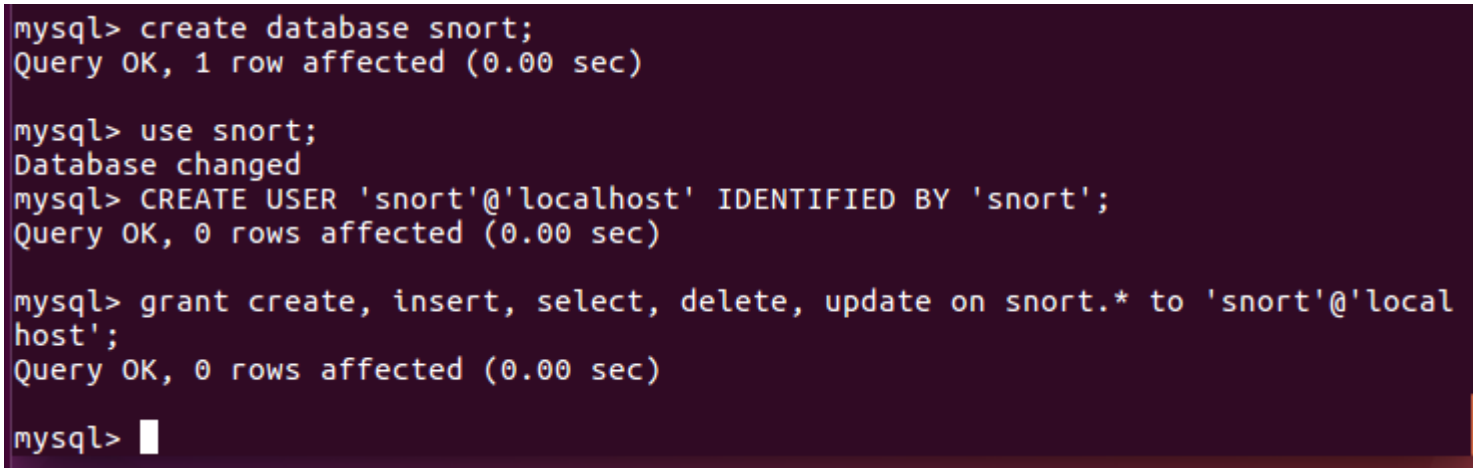

FIGURE 4.10 – Création de la base de donnée SNORT.

- Nous avons modifier le fichier de configuration de Barnyard2 **/etc/snort/barnyard2.conf** de renter les détails sur la base de donnée créé :
  sudo nano /etc/snort/barnyard2.conf
- nous avons ajouté les détails suivantes :
  output database : log, mysql, user=snort password=assil dbname=snort host=localhost
  Ici nous avons indié à snort qu'il faut enregistrer les événements dans une base de donnée mysql avec les paramètres ci-dessus. Le nom de la base de donnée **snort**, le nom d'utilisateur **snort** et le mot de passe **assil**.
- Le mot de passe se trouve dans le fichier Barnyard2, nous divrons empêcher les autres utilisateurs de le lire :
  sudo chmod o-r /etc/snort/barnyard2.conf
- Nous avons exécuté SNORT et Barnyard2 et généré les alertes :
  sudo /usr/local/bin/snort -q -u snort -g snort -c /etc/snort/snort.conf -i eth0 -D
- Pour dire à Barnyard2 de sauvegarder ces événements dans l'instance de la base de données de SNORT nous avons exécuté la commande suivante :
  sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log//snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort
- Enfin nous avons assuré si Barnyard2 a enregistré ces évènemets dans la base de données de SNORT avec la commande suivante :
  mysql -u snort -p -D snort -e "select count(*) from event"
- Le nombre d'évènements doit être supérieur à 0

4.5 Installation de BASE

- Nous avons installé d'abord les condition préalable :
  sudo add-apt-repository ppa :ondrej/php
  sudo apt-get update
  sudo apt-get install -y apache2 libapache2-mod-php7.0 php7.0 php7.0-mysql php7.0-common php7.0-gd php7.0-cli php-pear
  La figure 4.11 illustré l'insttalation des condition préalables de BASE.
- Ensuite nous avons téléchargé et installer ADODB :
  cd /snort—src

```
root@hanane-VirtualBox:~# sudo apt-get install -y apache2 libapache2-mod-php7.0
php7.0 php7.0-mysql php7.0-common php7.0-gd php7.0-cli php-pear
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.18-2ubuntu3.17).
libapache2-mod-php7.0 est déjà la version la plus récente (7.0.33-0ubuntu0.16.04
.16).
libapache2-mod-php7.0 passé en « installé manuellement ».
```

FIGURE 4.11 – Installation des conditions préalables de BASE.

wget https ://sourceforge.net/projects/adodb/files/adodb-php5-only/adodb-5.22/adodb-5.22.2.tar.gz

La figure 4.12 Téléchargement et l'installation ADODB.

```
root@hanane-VirtualBox:~/snort_src# wget http://sourceforge.net/projects/adodb/f
iles/adodb-php5-only/adodb-5.22/adodb-5.22.2.tar.gz
--2022-06-10 18:58:40--  http://sourceforge.net/projects/adodb/files/adodb-php5-
only/adodb-5.22/adodb-5.22.2.tar.gz
Résolution de sourceforge.net (sourceforge.net)… 104.18.34.243, 172.64.153.13
Connexion à sourceforge.net (sourceforge.net)|104.18.34.243|:80… connecté.
requête HTTP transmise, en attente de la réponse… 301 Moved Permanently
Emplacement : https://sourceforge.net/projects/adodb/files/adodb-php5-only/adodb
-5.22/adodb-5.22.2.tar.gz [suivant]
--2022-06-10 18:58:41--  https://sourceforge.net/projects/adodb/files/adodb-php5
-only/adodb-5.22/adodb-5.22.2.tar.gz
Connexion à sourceforge.net (sourceforge.net)|104.18.34.243|:443… connecté.
requête HTTP transmise, en attente de la réponse… 301 Moved Permanently
Emplacement : https://sourceforge.net/projects/adodb/files/adodb-php5-only/adodb
-5.22/adodb-5.22.2.tar.gz/ [suivant]
--2022-06-10 18:58:41--  https://sourceforge.net/projects/adodb/files/adodb-php5
```

FIGURE 4.12 – Télécharger et installer ADODB.

tar -xvzf adodb-5.22.2.tar.gz

sudo mv adodb5 /var/adodb

sudo chmod -R 755 /var/adodb

– nous avons téléchargé et installer BASE et copie-le à la racine apache2 :

cd /snort—src

wget

https ://sourceforge.net/projects/secureideas/files/BASE/base-1.4.5/base-1.4.5.tar.gz

Comme le montre la figure 4.13.

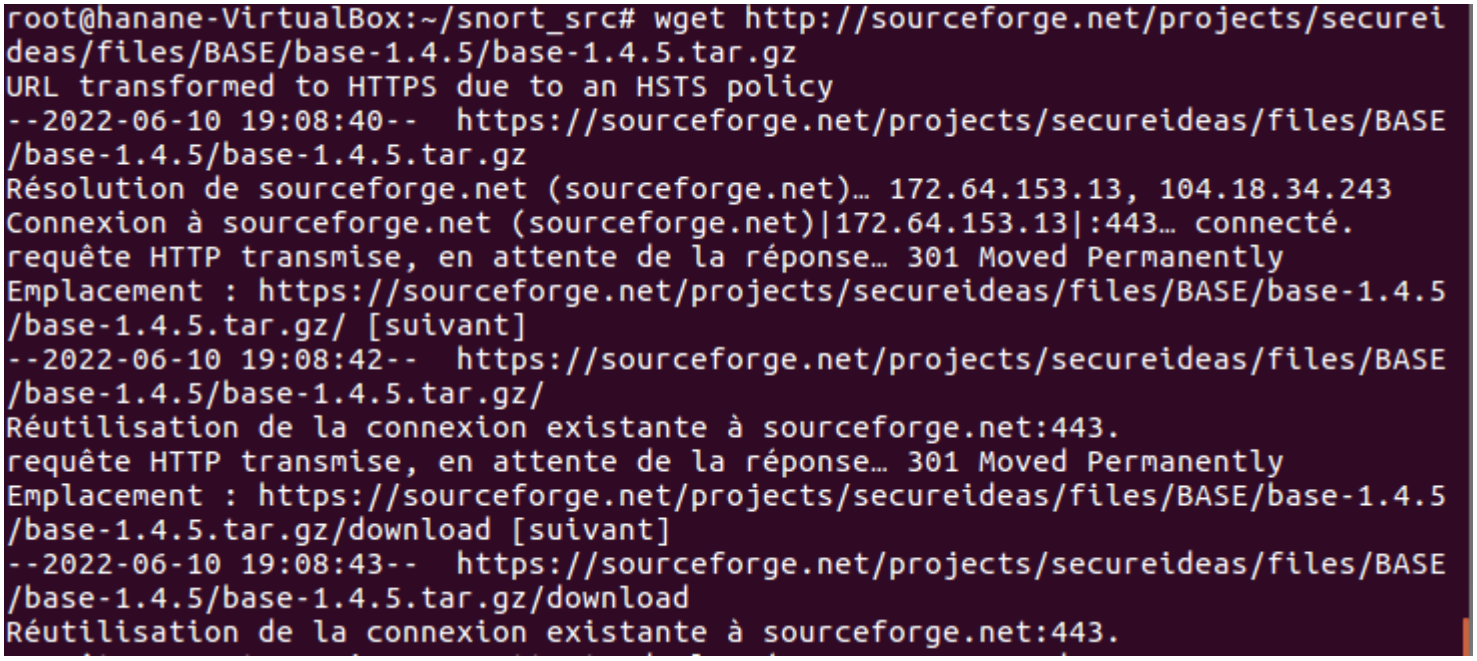

FIGURE 4.13 – Télécharger et installer BASE.

tar -xvzf base-1.4.5.tar.gz

sudo mv base-1.4.5 /var/www/html/base/

– Nous avons accédé au fichier **/var/www/html/base** par la commande suivante :

cd /var/www/html/base

– puis nous avons déplacé le fichier **base—conf.php.dist** vers **base—conf.php** avce la commande suivante :

sudo cp base—conf.php.dist base—conf.php

– ensuite nous avons changé des propriété et régler les autorisations : **sudo chwn -R**

www-data :www-data /var/www/html/base

sudo chmod o-r /var/www/html/base/base—conf.php

– Pour accéder et modifier le fichier de configuration **/var/www/html/base/base—conf.php**, nous avons utilisé les configurations nécessaires suivantes : **sudo pico**

/var/www/html/base/base—conf.php

Dans la ligne 50 **BASE—urlpath =’/base’;**

Dans la ligne 80 **DBlib—path =’/var/adodb/’;**

Dans la ligne 102 **alert—dbname =’snort’;**

Dans la ligne 103 **alert—host =’localhost’;**

Dans la ligne 104 **alert—port =”;**

Dans la ligne 105 **alert—user =’snort’;**

Dans la ligne 106 **alert—password=’liza2013’;**

– Enfin redémarrer le service apache2 :

sudo service apache2 restart

4.6 Lancement d’attaque

Pour faire le scénario d’attaque, nous avons utilisé deux PC, l’un joue le rôle de lanceur d’attaques et l’autre essaye de détecter l’attaque.

– Dans la machine victime nous avons lancé SNORT avec la commande suivante :

sudo snort -A console -i enp0s3 -u snort -g snort -c /etc/snort/snort.conf comme le montre la figure 4.14

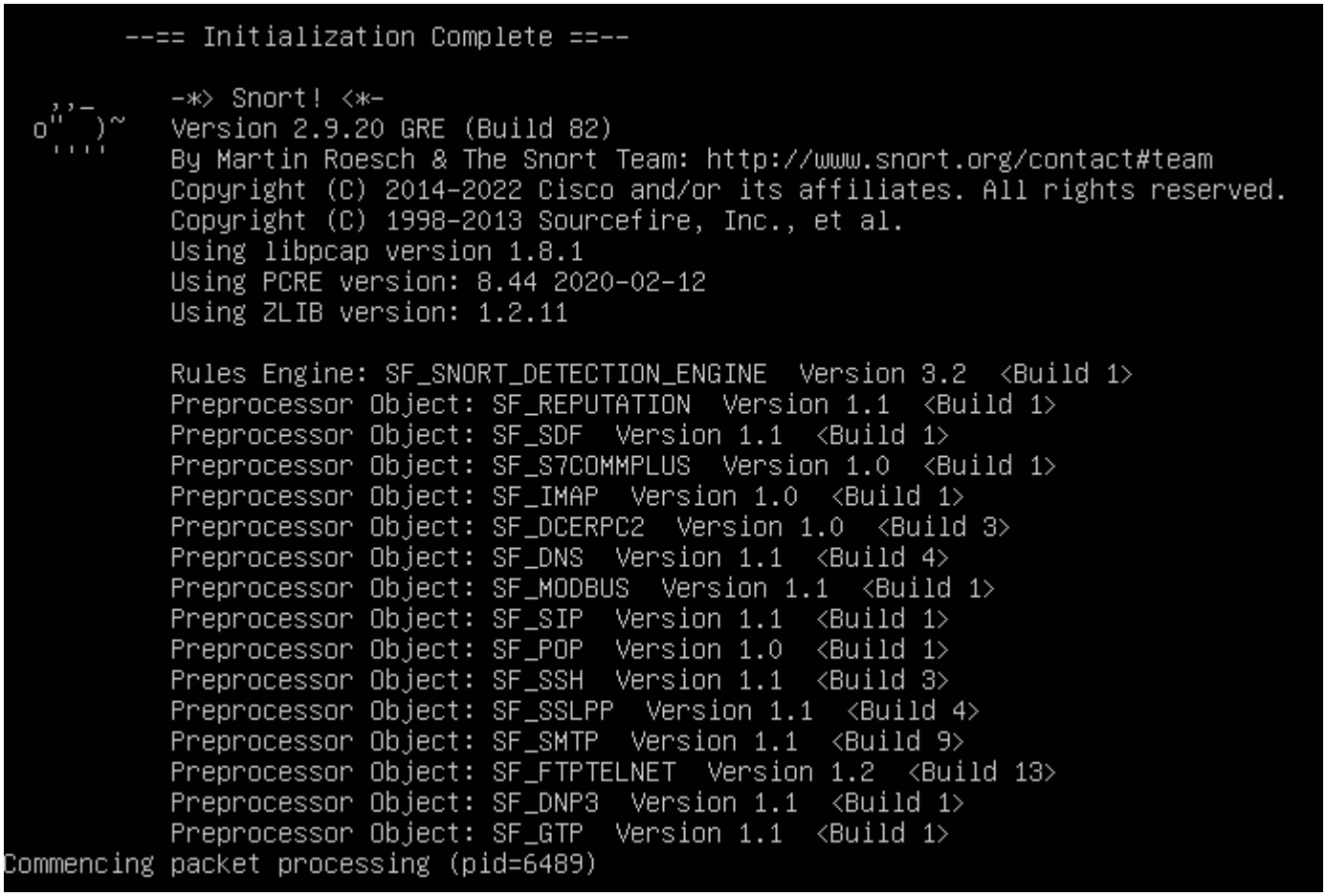

FIGURE 4.14 – Lancement de SNORT.

– puis nous avons lancer Barnyard2 :
sudo barnyard2 -c /etc/snort/barnyard2.conf -d /var/log/snort -f snort.u2 -w /var/log/snort/barnyard2.waldo -g snort -u snort
La figure 4.15 montre lancement de Barnyard2

```
database: compiled support for (mysql)
database: configured to use mysql
database: schema version = 107
database:           host = localhost
database:           user = snort
database:  database name = snort
database:    sensor name = hanane:NULL
database:      sensor id = 1
database:     sensor cid = 1
database:  data encoding = hex
database:   detail level = full
database:     ignore_bpf = no
database: using the "log" facility

       --== Initialization Complete ==--

  ______   -*> Barnyard2 <*-
 / ,,_  \  Version 2.1.14 (Build 337)
 |o"  )~|  By Ian Firns (SecurixLive): http://www.securixlive.com/
 + '''' +  (C) Copyright 2008-2013 Ian Firns <firnsy@securixlive.com>

WARNING: Ignoring corrupt/truncated waldofile '/var/log/snort/barnyard2.waldo'
Waiting for new spool file
```

FIGURE 4.15 – Lancement de Barnyard2.

– Après sa nous avons lancé un scan depuis la machine attaquante :
nmap -sF -v -O 192.168.120.140/24, comme le montre la figure 4.16

```
Nmap scan report for 192.168.120.243 [host down]
Nmap scan report for 192.168.120.244 [host down]
Nmap scan report for 192.168.120.245 [host down]
Nmap scan report for 192.168.120.246 [host down]
Nmap scan report for 192.168.120.247 [host down]
Nmap scan report for 192.168.120.248 [host down]
Nmap scan report for 192.168.120.249 [host down]
Nmap scan report for 192.168.120.250 [host down]
Nmap scan report for 192.168.120.251 [host down]
Nmap scan report for 192.168.120.252 [host down]
Nmap scan report for 192.168.120.253 [host down]
Nmap scan report for 192.168.120.254 [host down]
Nmap scan report for 192.168.120.255 [host down]
```

FIGURE 4.16 – Lancement d'attaque.

– Dans la machine victime nous avons remarqué rapidement la détection de ce scan depuis SNORT comme le montre la figure 4.17 :

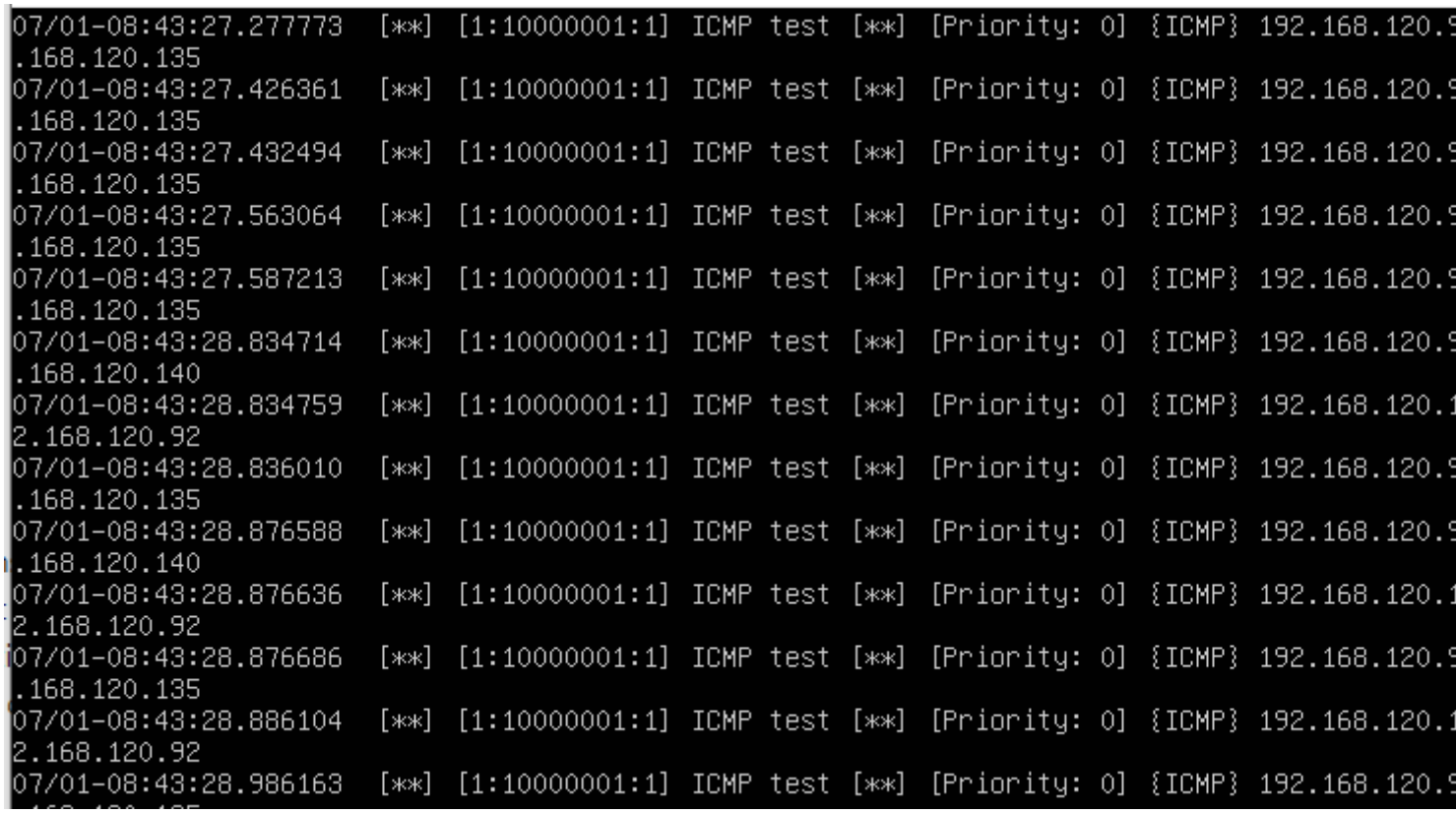

FIGURE 4.17 – Détection d'attaque.

conclusion

Dans ce chapitre nous avons réalisé l'installation et la configuration de SNORT, avec l'installation des dépendences de snort Barnyard2 et BASE qui sont nécessaire pour le fonctionnement de SNORT. Nous avons vu à la fin, comment Snort a pu stopper une attaque avec succès.

Conclusion générale

Dans cet mémoire, nous avons mis en place un système de détection d'intrusion à l'aide de SNORT. Cette stratégie de sécurité efficace pour les réseaux informatiques.
Dans un premier temps, nous avons donné un aperçu sur les systèmes de détection d'intrusion, et nous avons présenté quelques généralités l'IDS. Ensuite, nous avons opté par SNORT, car c'est un très bon outil. Le logiciel Snort est un système de détection au niveau réseau (NIDS). Dans une première étape, nous avons installé et configuré snort, puis nous avons configuré sa base de données afin d'organiser méthodiquement les alertes. Dans une deuxième étape, nous avons réalisé un test d'intrusion en utilisant deux machines l'une pour lancé une attaque scan réseau et l'autre pour détecter cette dernière.
Le résultat du test que nous avons utilisé est satisfaisant, mais cela ne signifie pas que le système est entièrement fonctionnel car aucun système de sécurité ne peut garantir une sécurité absolument fiable.
Perspectives
– Nous envisageons de tester notre syqtème en lançant d'autres types d'attaques notamment DDOS.
– Nous prévoyons des mises à jours quotidiennes pour se prévenir contre les nouvelles attaques.

Bibliographie

[1] Thierry Evangelista. *Les système de détection d'intrusions informatique*. 2004.

[2] Abdelhalim ZAIDI. *Recherche et détection des patterns d'attaques dans les réseaux IP à haut débits*. 2011.

[3] SELMANI Elgharbi. *Mise en place d'un IDS pour sécuriser un réseau en utilisant Snort*. 2020.

[4] BEN BRAHIM Embarka et AMICHE Selyna. *Mise en place d'une solution de détection d'intrusion*. 2017.

[5] DABOUR Imane et HADJI Imène. *Etude et mise en place d'un système de détection/prévention d'intrusion (IDS/IPS) réseau, Etude de cas SNORT*. 2014.

[6] https ://lgm.univ mlv.fr. *HIDS et systèmes Unix-IGM*. juin, 2022.

[7] Raffeq rehman. *Intrusion detection systems with snort advanced IDS techniques using snort, apache, mysql, php, and ACID*. 2003.

[8] IBRAHIM Mohamed Amine et TEBOURKI Hamdi. *Installation et configuration d'un système de détection d'intrusion (IDS)*. 2009.

[9] Thomas ORY. *Articles pae data*. 2020.

[10] Mécanisme HIDS et système unix IGM. *htt ://igm.univ-mlv.fr*. 2022.

[11] DURAND Sébastien. *Systéme de détection d'intrusion SNORT*. 2003.

[12] Llir KADRIU. *Utilisation de SNORT dans une PME*. 2019.

[13] FATHI Ben Nasr et Alia Khessairi ABBASSI. *Mise en place d'une sonde SNORT*. 2004.

[14] TOUATI Azedine. *Détection d'intrusions dans les réseaux LAN : installation et configuration de l'IDS-SNORT*. 2016.

[15] HAMZATA Gueye. *Mise en place d'un ids en utilisant SNORT*. 2010.

[16] Host based IDS with SNORT. *Barnyard2 ans snorby in AWS*. juin, 2022.

[17] Lamia MOUZAIA et Riadh AMOURAT. *Etude et amélioration de la chaine logistique agro-alimentaire du groupe cevital*. 2020.

[18] Groupe CEVITAL. *https ://www.cevital.com*. juin, 2022.

[19] AZIROU Millissa et FAID Amine. *Mise en place d'un référentiel de sécurité cas d'étude : group CEVITAL*. 2018.

[20] Virtualbox. *https ://www.blogdumoderateur.com/tools/oracle-vm-virtualbox*. juin, 2022.

[21] Ubuntu. *https ://ubuntu.com*. juin, 2022.

[22] Data acquisition (DAQ). *https ://ajolly.fr*. juin, 2022.

[23] SNORT. *https ://www.snort.org*. juin, 2022.

[24] Installing barnyard2. *https ://sublimerobots.com*. juin, 2022.

[25] How to install snort on ubuntu. *https ://upcloud.com.* juin, 2022.

[26] Installing BASE on ubuntu. *https ://sublimerobots.com.* juin, 2022.

Résumé

Les réseaux informatiques sont devenus les plus vulnérables, dont diverses attaques qui sont mises en place par des outils d'évaluation, il est donc devenu nécessaire de détecter les intrusions et ces attaques lorsqu'elles surviennent, et cela est devenu possible grâce aux mécanismes de détection d'intrusion.
Nous avons vu, la sécurisation d'un réseau est une étape délicate permettant de protéger une entreprise des risques les plus courants,émanants aussi bien de l'internet que de son propre réseau local.
Notre projet vise à étudier et développer un système permettant de protéger et de maintenir le réseau informatique, qui s'appelle le système de détection d'intrusion(ids), qui sont des outils permettant de détecter les attaques/intrusions du réseau sur lequel il est placé. C'est un outil complémentaire aux firewall, scanneurs de failles et anti virus, et qui dépend de SNORT, qui consiste à un ensemble de règles qui à leur tour sont basées sur l'alerte et l'envoi de signaux par Être attaqué.
La détection d'intrusion consiste à trouver des systèmes informatiques utilisés à des fins illicites, tandis que les systèmes de prévention d'intrusion tentent de les arrêter en notifiant les pare-feu des tentatives d'intrusion.
Mots-clés : Réseau, informatique, IDS, attaque, intrusion, firewells, règles, SNORT.

Abstract

Computer networks have become the most vulnerable, including various attacks that are implemented by assessment tools, so it has become necessary to detect intrusions and these attacks when they occur, and this has become possible thanks to the mechanisms of intrusion detection.
We have seen that securing a network is a delicate step in protecting a company from the most common risks, emanating both from the Internet and from its own local network.
Our project aims to study and develop a system to protect and maintain the computer network, which is called the intrusion detection system (ids), which are tools to detect attacks / intrusions of the network on which it is placed. It is a complementary tool to firewalls, fault scanners and anti-viruses, and which depends on SNORT, which consists of a set of rules which in turn are based on alerting and sending signals by Being Attacked.
Intrusion detection involves finding computer systems that are being used for illicit purposes, while intrusion prevention systems attempt to stop them by notifying firewalls of intrusion attempts.
Key words : network, informatique, IDS, attacks, intrusion, firwells, rules, SNORT .